



Universidad
Carlos III de Madrid

Ingeniería Técnica en Informática de Gestión

PROYECTO FIN DE CARRERA

NON STOP: Diseño e implementación de
aplicaciones colaborativas multi-dispositivo en el
entorno de la práctica deportiva

Autor: Alberto Herranz Martín

Tutor: Jose María Álvarez Rodríguez

Leganés, octubre de 2015

Agradecimientos.

“No dejes para mañana lo que puedes hacer hoy”. Esta es la conclusión y resumen de los últimos años y meses en los que nunca me decidí a realizar el proyecto fin de carrera, pero finalmente me lo propuse y lo conseguí, pero todo hay que decirlo, casi a última hora.

Dar las gracias a mi tutor, por orientarme y animarme durante la realización del proyecto.

Dar las gracias a mi familia, a mi novia y a mis amigos, por apoyarme y entenderme durante estos días en los que he estado más ausente.

Finalmente, dedicar este trabajo a mi padre, sin duda, la persona que más se va a alegrar con la finalización del proyecto fin de carrera.

Resumen.

El objetivo principal del proyecto fin de carrera “*Non-Stop*” es diseñar e implementar un sistema de aplicaciones colaborativas que permita trabajar de manera conjunta a cada una de ellas. Este sistema es aplicado a la práctica deportiva y su funcionamiento se inicia con la emisión de una señal de localización GPS desde un dispositivo móvil. La señal es recibida y analizada por el sistema y se generará una alarma en los casos en los que se detecte una parada inesperada. El diseño del sistema se basa en una arquitectura e implementación sencilla pero de gran utilidad para un colectivo de personas en aumento, ciclistas, corredores, etc.

Abstract.

The main objective of the project "Non-Stop" is design and implement a core set of collaborative applications that allow to work together for each one. This system would be applied in sport practice, the procedure starts with the emission of a location GPS signal since a mobile device. The signal is received and analyzed by the system, an alarm will be generate whenever the system detect a unexpected shutdown. The design is based on a simple architecture and implementation but very useful for a increasing group of persons: runners, cyclist...

Índice general

Capítulo 1. Introducción	1
1.1. Propósito y audiencia	1
Capítulo 2. Problemática y necesidad	2
2.1. Situación actual	2
2.2. Problemática y necesidad	2
2.3. Motivación y objetivos	4
Capítulo 3. Estado del arte	7
3.1. Aplicaciones y sistemas similares	8
3.2. Java	11
3.3. Eclipse	13
3.4. Oracle	16
3.5. Apache Tomcat 7	18
3.6. GPS	19
3.7. Formula Haversine	22
3.8. API – Google Maps	24
Capítulo 4. Análisis	25
4.1. Definición de requisitos	26
4.1.1. Definición de requisitos funcionales	29
4.1.2. Definición de requisitos no funcionales	40
4.2. Escenarios y casos de uso	43
4.2.1. Actores	44
4.2.2. Representación gráfica casos de uso	45
4.5. Matriz de trazabilidad casos de uso y requisitos funcionales	47
Capítulo 5. Diseño	48
5.1. Arquitectura del sistema	49
5.2. Estructura lógica del modelo	51
5.3. Estructura física del modelo	54
5.4. Diagrama de secuencia	57

Capítulo 6. Implementación	58
6.1. Clases de dominio.....	59
6.2. Clases de servicio	60
 Capítulo 7. Pruebas y validación.....	 61
7.1. Caso de prueba 1	62
7.2. Caso de prueba 2	64
7.3. Caso de prueba 3	66
7.4. Caso de prueba 4	68
7.5. Caso de prueba 5	68
 Capítulo 8. Planificación y presupuesto	 72
8.1. Planificación del proyecto	72
8.2. Presupuesto del proyecto	74
8.2.1. Costes de recursos humanos	74
8.2.2. Costes de material.....	75
8.2.3. Costes indirectos.....	76
8.2.4. Resumen de costes.....	76
 Capítulo 9. Conclusiones y trabajo futuro	 77
 Capítulo 10. Bibliografía y referencias	 81
 Apéndices.....	 82

Capítulo 1. Introducción

1.1. Propósito y audiencia

El presente documento hace referencia al Proyecto fin de carrera (PFC), “*Non-Stop*”, cuyo objetivo es diseñar y desarrollar un sistema que permita interconectar diferentes dispositivos, móvil y ordenador personal (PC), a través de un componente principal integrado en un servidor. El sistema se inicia con la emisión de una señal de localización de posicionamiento global (GPS) desde un dispositivo móvil que es recibida y analizada por el sistema principal, permitiendo que las aplicaciones integradas en los diferentes dispositivos funcionen e interactúen de manera conjunta.

El sistema cuenta con una base de datos (BBDD) de apoyo que permite la recepción de las diferentes señales de localización, aportando mayor eficiencia en el acceso posterior a la información e independencia de la aplicación principal y el software asociado.

Una vez descrito a grandes rasgos el propósito del sistema, se describe brevemente la funcionalidad del proyecto, “*Non-Stop*”. El PFC pretende aportar una serie de funcionalidades para corredores, ciclistas y otros deportistas que realizan la mayor parte de sus actividades al aire libre y puedan estar localizados y monitorizados en todo momento desde cualquier otro terminal móvil o PC por parte de familiares o amigos.

La tercera parte del proyecto es el diseño de un algoritmo cuyo objetivo es analizar la recepción de las diferentes señales de localización GPS recibidas, el algoritmo determina si se ha producido una parada no esperada durante una actividad deportiva. Sí un deportista que se encuentra realizando una actividad al aire libre está parado, normalmente es sinónimo de un problema durante la realización de la actividad (caída, indisposición física, etc). De esta forma, se dota al sistema de un componente funcional de gran utilidad a la localización GPS, con un sistema de alarma que avisa a las personas interesadas de un posible accidente.

Resumiendo, el proyecto se estructura en 3 partes.

- Sistema servidor. Receptor de señal GPS y emisor de alarma, almacena la información en BBDD para la explotación de ésta en un momento posterior o de manera on-line.
- Sistema móvil. Emisor de señal GPS.
- Algoritmo. Sistema de alarma, ubicado en el sistema servidor.

Capítulo 2. Problemática y necesidad

2.1. Situación actual

El PFC analiza la situación actual y valora dos aspectos importantes, la era de las nuevas tecnologías y de la práctica deportiva de grandes masas. El número de usuarios de internet y dispositivos móviles sigue en aumento, siendo el teléfono móvil el dispositivo electrónico más utilizado en España para conectarse a la Red.

Basándonos en las estadísticas e información que proporciona el Instituto Nacional de Estadística (INE) [REF 1] en una de sus notas de prensa:

“El 74,4% de los hogares dispone de conexión a Internet, casi cinco puntos más que en 2013.

Por primera vez en España hay más usuarios de Internet (76,2%) que de ordenador(73,3%). El 77,1% de los internautas accedieron a Internet mediante el teléfono móvil.”

Evolución del equipamiento TIC en los hogares
Serie homogénea 2006-2014. Total nacional (% de hogares)

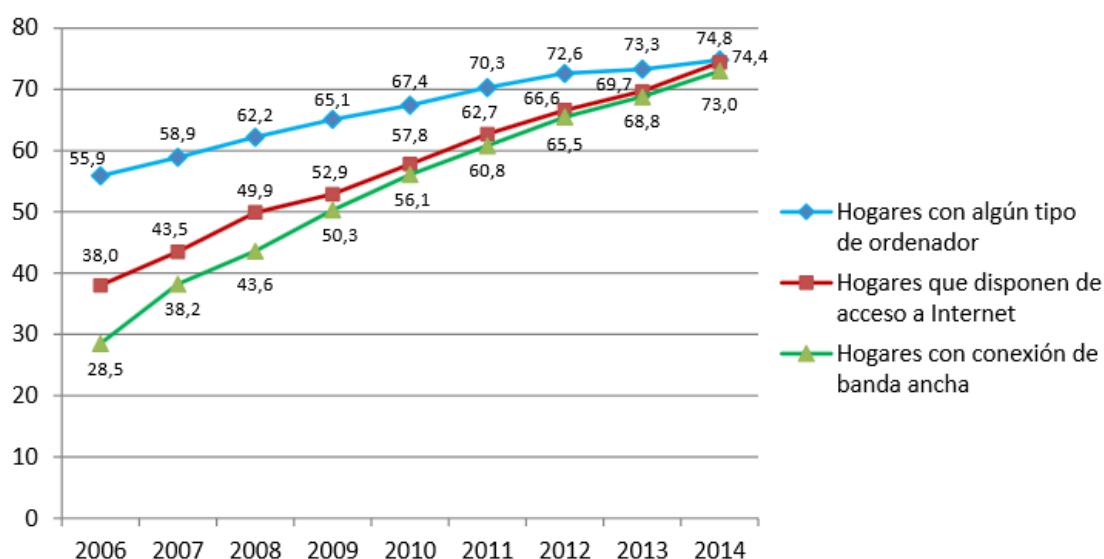


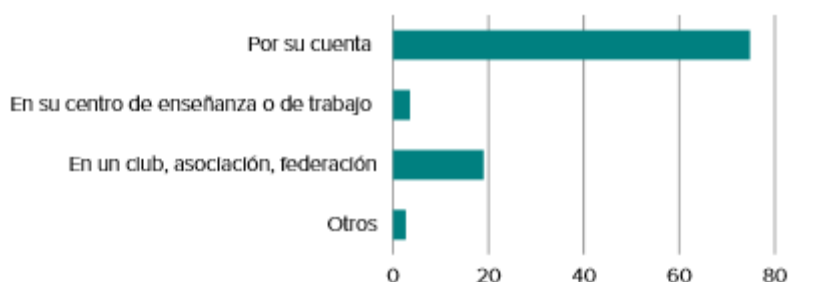
Figura 2.1.1 – Evolución TIC en Hogares.

Fuente: <http://www.ine.es/>

Como se puede ver en la gráfica de la figura 2.1.1, la evolución en el uso del teléfono móvil e Internet sigue una tendencia alcista, siendo datos proporcionados por el INE para el año 2014 que previsiblemente irán en aumento en los próximos años.

Por otro lado, según estadísticas de Consejo Superior de Deportes (CSD) [REF 2] el número de personas que realizan actividades deportivas por su cuenta y al aire libre también es elevado.

“Uno de los elementos que representan de manera más gráfica el desarrollo cultural y social de la población española en las últimas tres o cuatro décadas – es decir, desde que nuestro país adquiere la condición de un estado democrático- es el nivel de práctica deportiva de la misma”



Fuente: Consejo Superior de Deportes. Encuesta de Hábitos Deportivos en España

Figura 2.1.2 – Personas de más de 15 años que practican deporte según forma organizativa.

Fuente: www.csd.gob.es

El desarrollo de actividades deportivas por cuenta propia supone, en cierto modo, un factor de riesgo o grado de incertidumbre en el entorno familiar que nos rodea. Son muchas las horas que pueden pasarse fuera de casa, y en las cuales se desconoce la ubicación y situación en la que nos encontramos.

Siendo esta la situación actual, ¿por qué no encontrar un punto de unión entre el potencial de las nuevas tecnologías y la práctica de deporte? Como respuesta a esta pregunta surge la idea del PFC, crear un sistema que permita monitorizar la práctica deportiva en exteriores y realizar un seguimiento o enviar un aviso en caso de detectar una parada por parte del deportista, de ahí el nombre del proyecto “Non-Stop”.

2.2. Problemática y necesidad

Teniendo en cuenta la situación actual y el incremento de actividades deportivas al aire libre por parte de un colectivo mayoritario, deportistas por cuenta propia, surge la necesidad de crear una aplicación que permita monitorizar en tiempo real la localización del deportista y generar un sistema de alarmas en caso de no detectarse movimiento entre diferentes señales de localización.

El potencial de la aplicación es enorme. La fase inicial se basa en el diseño del sistema y definición del algoritmo de emisión de alarmas, pero con las bases iniciales se permitirá incluir nuevas funcionalidades y evoluciones en el sistema:

- Envío de señal si la batería del móvil empieza a ser reducida evitándose falsos positivos.
- Planificación de rutas y puntos de control en función de una hora prevista, con su correspondiente envío de señal en caso de no cubrir el sector correspondiente en el tiempo esperado.
- Envío de mensajes cortos (SMS) a grupos.

No obstante, en esta primera fase de proyecto, el alcance inicial es crear un sistema de alarmas que permita visualizar por pantalla la situación GPS actual, basándonos en un sistema cliente – servidor que se detallará en secciones posteriores.

2.3. Motivación y objetivos

En primer lugar, indicar que la motivación del autor, una persona deportista que se encuentra dentro del colectivo descrito anteriormente, es la creación de un sistema de alarmas automatizado que permita su localización en caso de emergencia. Un deportista que pasa muchas horas al aire libre, realizando actividades como correr o montar en bicicleta, en la mayoría de las ocasiones las realiza en solitario. Esto puede generar situaciones inesperadas de riesgo, como caídas, problemas mecánicos en bicicleta o cualquier otra indisposición o complicación.

Al iniciar una actividad deportiva, se formulan las siguientes preguntas por parte de familiares, ¿por dónde va a salir a entrenar?, ¿a qué hora va a llegar?, ¿si le pasa algo, cómo me entero?, etc.

Las respuestas en la mayoría de las ocasiones carecen de objetividad, porque las situaciones inesperadas que se producen se escapan del control del deportista. Esta es la motivación por la que se trata de una idea innovadora para el PFC. El diseño de una aplicación que permite realizar seguimiento de una práctica deportiva y una parte más importante, generar un sistema automatizado de alarmas y avisos en caso de parada inesperada.

El objetivo, desde el punto de vista tecnológico, no es otro que desarrollar una aplicación cliente-servidor basado en software libre que permita desarrollar un software que pueda evolucionar en el tiempo con nuevas funcionalidades según las necesidades de usuarios.

Identificadas las necesidades de usuario, como es la generación automática en caso de emergencia o parada inesperada, el objetivo del sistema es posibilitar la localización del usuario e identificar y analizar su posición para determinar si se ha producido una parada sin necesidad que el usuario interactúe con la aplicación. Por lo tanto, el sistema debe cumplir los siguientes objetivos.

- Iniciar la aplicación desde dispositivo móvil. El dispositivo móvil es el emisor de la señal GPS. Esta es la única acción requerida para que se inicie el sistema y empiece a analizar la recepción de señales.
- Recepción de señal. El sistema principal, recibe la señal GPS y analiza las diferentes localizaciones emitidas desde el dispositivo móvil.
- Generación de alarmas y almacenamiento de la ruta. El sistema principal, debe almacenar la ruta y en el caso que el algoritmo detecte una parada, generar una alarma y guardarla.
- El sistema, si identifica una alarma, realiza el envío de un SMS a los destinatarios registrados.
- Los usuarios pueden acceder a la aplicación principal, permitiendo ver la ubicación actual del usuario móvil y las alarmas generadas, independientemente de haber recibido un SMS.

Con este sistema, se permite tener monitorizado al usuario a través de su última localización y el mapa dibujado durante su actividad y generar una alarma y aviso en caso de parada con su localización. La localización asociada a la alarma permite en el caso de situaciones extremas realizar una búsqueda y evacuación.

Capítulo 3. Estado del arte

En el siguiente apartado se describe en líneas generales el establecimiento de las bases para la creación del sistema de forma rápida y eficiente así como el análisis de otras aplicaciones similares a la idea del PFC.

Desde el punto de vista de la tecnología, se debe conseguir:

- Ofrecer tiempos de desarrollo ajustados, junto a un software de calidad y de fácil mantenimiento.
- Orientación *Open Source* (Código abierto). Basado en la filosofía de recoger nuevas funcionalidades y otros conocimientos que ofrece la comunidad y tener una participación activa en ella.

Para poder cumplir los objetivos establecidos, la aplicación debe apoyarse en principios de desarrollo de aplicaciones ágil, garantizándose que el software pueda seguir ejecutándose y evolucionando de forma segura y eficaz con el paso del tiempo. La aplicación no puede quedar obsoleta desde el punto de vista de la tecnología aplicada y de las nuevas funcionalidades y evoluciones aportadas por la comunidad.

La aplicación principal, deberá contar con otros elementos importantes como:

- Acceso a base de datos y persistencia.
- Servidor Web.

Una vez se tiene una visión global de la diferentes tecnologías a utilizar se pasará a analizar de forma detallada cada una de ellas, puntualizándose los principales beneficios que aportarán a nuestro proyecto y teniendo en cuenta que todas las aplicaciones en las que se basa la arquitectura del PFC es de carácter gratuito y código libre.

Por último, se desarrolla dentro del apartado “Estado del Arte” las principales características de otros elementos necesarios dentro de la definición del sistema del proyecto PFC, como es el sistema de localización GPS, Formula Haversine.

3.1. Aplicaciones y sistemas similares

Cómo se ha detallado en el capítulo 2 de este documento con estadísticas aportadas por el INE y el CSD, vivimos en la era de las nuevas tecnologías y de la práctica deportiva de grandes masas.

El número de usuarios de internet y dispositivos móviles alcanza máximos históricos y la práctica deportiva en el exterior se incrementa a pasos agigantados; por lo tanto, son muchas las empresas que quieren explotar este sector con aplicaciones similares a la propuesta en el PFC que nos ocupa, convirtiéndose en un mercado muy atractivo con un número importante de clientes potenciales.

Las primeras aplicaciones existentes de monitorización requerían de un dispositivo externo, con una antena GPS integrada, encargado de la recepción de la información y una posterior conexión con un PC para registrar los datos, visualizarlos y analizarlos. Posteriormente nacieron aplicaciones integradas en teléfonos móviles, dado que estos dispositivos evolucionaron notablemente convirtiéndose en teléfonos inteligentes con antenas GPS dentro de los terminales móviles. La ventaja principal aportada por estas evoluciones tecnológicas es la posibilidad de realizar una monitorización de la actividad de forma on-line.



Figura 3.1.1 – Receptor GPS Tradicional.

Fuente: https://es.wikipedia.org/wiki/Sistema_de_posicionamiento_global

En la actualidad, son muchas las aplicaciones con una funcionalidad similar a la idea del PFC, permiten realizar un registro de las actividades deportivas realizadas para tener un control y realizar seguimiento de entrenamientos y avances en el rendimiento, pero surge una pregunta importante.

¿Están enfocadas estas aplicaciones desde el punto de vista de alarma o gestión de emergencias? En principio, todas ellas permiten un seguimiento on-line de la actividad registrada, basándose en los avances tecnológicos introducidos en los teléfonos móviles, pero no se gestionan avisos por problemas o paradas imprevistas, por lo que encontramos un elemento diferenciador entre estas aplicaciones y el sistema “Non-Stop”.

Se procede a detallar algunas de estas aplicaciones con su descripción comercial e información aportada desde su propia página. Se podrían incluir muchas más aplicaciones, que ratifica que se trata de una línea tecnológica atractiva y en constante evolución, pero se considera que la muestra aportada es representativa de la situación actual sobre aplicaciones similares.

- Endomondo [REF 12] – *“Es una aplicación gratuita y página web que hace que el ejercicio físico sea más divertido y te motive a continuar”*
- Runtastic [REF 13] – *“Aplicación personal para todos los deportes al aire libre(correr, caminar, senderismo, ciclismo...) y en interior(cinta, cardio, yoga, entrenamiento de fuerza). Registra tus actividades, incrementa tu motivación, quema más calorías, logra mejores resultados y supera tus metas”*
- Garmin Connect [REF 14] – *“Conecta con otros. Seguir, analizar, compartir y recibir ánimo en Garmin Connect.”*

Todas estas aplicaciones funcionan de la misma forma, recepción de señal GPS desde dispositivo móvil y monitorización desde sistema principal de la actividad realizada. La diferencia entre las dos primeras aplicaciones y Garmin Connect es que Garmin Connect requiere de un dispositivo adicional para recibir la señal GPS y una posterior conexión al dispositivo móvil, para transferir la información al sistema, sistema tradicional de monitorización de actividades deportivas. Este sistema supone un encarecimiento en el producto, dado que se requiere de un dispositivo adicional con un coste importante, aunque supone beneficios como son dispositivos más pequeños e independencia de una tarifa telefónica o datos.

Todas estas aplicaciones, además de proporcionar una versión gratuita, también disponen de opciones “Premium” o “Pro” dotadas de funcionalidades exclusivas y más avanzadas.

El sistema de localización ‘Non-Stop’ ofrece la misma funcionalidad, con el aporte adicional y de gran utilidad cómo es la gestión de alarmas que hace que la diferente al resto.

Por último analizar otra aplicación, en las últimas fechas ha aparecido un nuevo sistema, con una funcionalidad diferente del resto, incluso con publicidad en TV, lo que demuestra la importancia que están dando las empresas a este sector comercial, cuyo eslogan es: “La aplicación que te puede salvar la vida”.

Alpify [REF 15]. Se trata de una aplicación para dispositivos “Smartphone” con conexión de datos que permite enviar la geo-posición o coordenadas exactas del usuario en el momento en que este realiza una llamada al 112. Conocer dónde está exactamente la persona que está pidiendo socorro es una información de gran importancia para organizar y agilizar el rescate, y con ello se podrán optimizar medios y acortar tiempos de búsqueda.

¿Se está hablando de la misma aplicación? En un principio su definición se ajusta a la idea del PFC. Esto supone una pequeña decepción, pero también puede aportar soluciones o funcionalidades no contempladas.

Se pasa a valorar y analizar esta aplicación con un detalle mayor que las anteriores, dado que ofrece una funcionalidad similar a la propuesta en la aplicación ‘Non-Stop’, en búsqueda de características diferenciadoras.

La aplicación puede activarse en dos modos:

- Activándola cuando se necesita llamar al 112: pulsando en el botón de emergencia de la aplicación el teléfono buscará conexión de datos, o en su defecto conexión de telefonía móvil (la que el cliente tenga contratada), y enviará, vía datos o de SMS, las coordenadas de posición de la llamada al centro 112.
- Activándola cuando se inicia la excursión: de esta manera el sistema registra un “Track” o recorrido del usuario mientras este tenga cobertura de datos, y si en un momento dado la pierde siempre podrá saberse cuál era su posición antes de perder cobertura.

El sistema de alarmas “Non-Stop” ofrece una fusión de las funcionalidades aportadas por esta aplicación, con las siguientes diferencias y ventajas:

- Aporta un algoritmo que controla si el avance en la actividad deportiva sigue su curso y no requiere de activación por parte del usuario. Si un usuario sufre un accidente, puede darse el caso que no pueda activar el botón para solicitar ayuda porque pudiera estar indispuerto. Por lo tanto, aporta automatismo en el aviso.
- El envío de SMS no se realiza al servicio de emergencias 112, sino a la persona o personas de contacto que se indiquen. Se debe evitar saturar este sistema en casos de falsos positivos. Se considera que el aviso debe ser recibido por familiares y en el caso de ser necesario que sean ellos quien gestionen con el 112 la evacuación en caso de situaciones extremas.
- La monitorización remota del Track, ofrece la misma funcionalidad, con la opción de generarla de forma privada. Aunque no debería ser la opción a utilizar, permite dotar de privacidad a los usuarios que lo consideren.

Por lo tanto, el sistema “Non-Stop” incluye los principios de monitorización clásico y el sistema de alarmas automatizado que no incluye la aplicación Alpyfi, abriéndose una línea de trabajo interesante para los sistemas de monitorización.

3.2. Java

Se formulan varias preguntas para definir los principales beneficios aportados por Java y el motivo por el que es elegido como tecnología y lenguaje de programación de muchas aplicaciones y sistemas.

¿Qué es la tecnología Java y para qué es útil?

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tengan Java instalado y cada día es mayor el número de aplicaciones y sistemas que se crean con esta tecnología. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes. Java es rápido, seguro y fiable.

¿La descarga de Java es gratuita?

Sí, la descarga de Java es gratuita. Puede obtener la última versión en la página oficial de Java [REF 4]

¿Es fácil estar actualizado?

¿Por qué debería actualizarme a la versión más reciente de Java?

La versión más reciente de Java contiene importantes mejoras para el rendimiento, estabilidad y seguridad de las aplicaciones Java que se ejecutan en su equipo. La instalación de la actualización gratuita garantiza que sus aplicaciones Java sigan ejecutándose de forma segura y eficaz.

Resumiendo, las principales características que aporta Java son:

- Rápido, seguro y fiable.
- Gratuito.
- Multiplataforma y portable.
- Actualización continua.

Al realizar la descarga, se obtiene el software Java Runtime Environment (JRE). JRE está formado por Java Virtual Machine (JVM), clases del núcleo de la plataforma Java y bibliotecas de la plataforma Java de soporte. JRE es la parte de tiempo de ejecución del software de Java, que es todo lo que necesita para ejecutarlo en el explorador web.

Por último, la escritura de aplicaciones y *Applets* de Java necesita herramientas de desarrollo como *Java Development Kit* (JDK). JDK incluye *Java Runtime Environment*, el compilador Java y el interface de programación de aplicaciones (API) de Java.

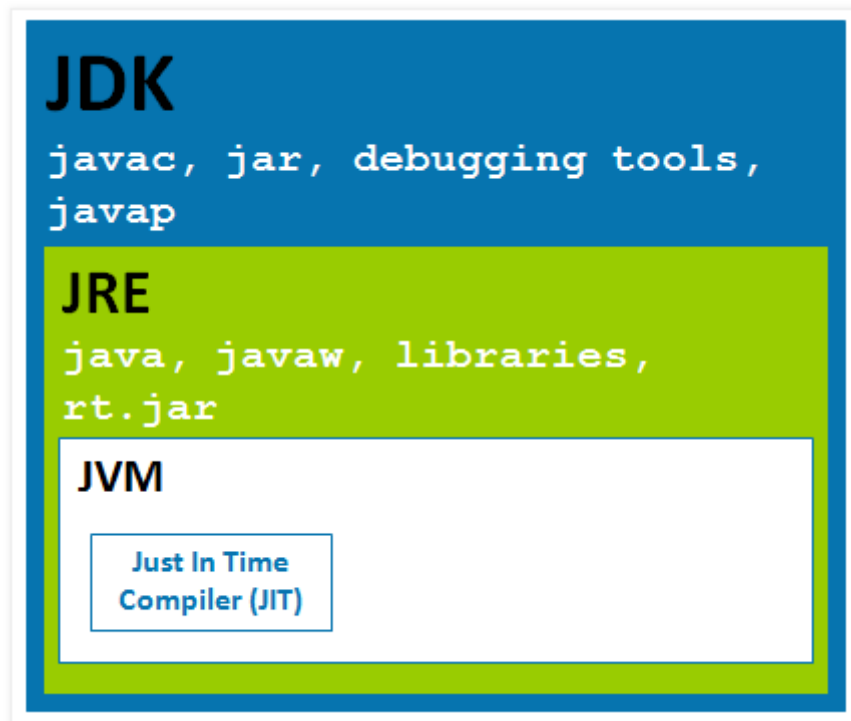


Figura 3.2.1 –Java Development Kit.

Fuente: <http://www.javabeat.net>

Para la ejecución del proyecto se instala la versión JDK 1.8 descargada desde la url - <https://www.java.com/es>. Una vez descargada, se puede proceder a verificar su correcta instalación en el entorno de trabajo.

```
C:\>java -version
java version "1.8.0_45"
Java(TM) SE Runtime Environment (build 1.8.0_45-b15)
Java HotSpot(TM) Client VM (build 25.45-b02, mixed mode, sharing)
C:\>
```

Figura 3.2.2 – Instalación local versión Java

3.3. Eclipse

Eclipse es un Entorno de Desarrollo Integrado (en inglés IDE) de código abierto, especializado en Java y desarrollado originalmente por IBM [REF 7], aunque en la actualidad pertenece a la fundación Eclipse [REF 3], una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación; es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Provee al programador de un entorno de trabajo amigable en el que desarrollar aplicaciones de manera más sencilla y con menor tendencia a cometer errores.

Aunque Eclipse pueda ser utilizado con múltiples lenguajes de programación, está implementado para ser usado principalmente con Java (apartado 3.2) y a la hora de descargarlo, incluye todas las herramientas para programar en este lenguaje. Algunas de las funciones de mayor utilidad a la hora de desarrollar aplicaciones Java son:

- Sistema de compilación en tiempo real que detecta los errores sintácticos y semánticos sin necesidad de usar un proceso de compilación estándar.
- Sistema de ayuda y sugerencias en el que se muestran los métodos pertenecientes a un objeto y descripción de cada método.
- Ejecución de las aplicaciones rápida desde el entorno de desarrollo.

Eclipse es uno de los entornos Java más utilizados a nivel profesional y el paquete básico se puede expandir mediante la instalación de plugins para añadir nuevas funcionalidades a medida que se vayan necesitando. Por lo tanto, Eclipse ofrece la posibilidad de aumentar su funcionalidad a base de plugins desarrollados por la comunidad de usuarios o por empresas. Estos plugins se pueden instalar manualmente usando la herramienta integrada dentro de Eclipse para agregar nuevo software – Eclipse Marketplace.

Eclipse Marketplace, es un gran repositorio web integrado dentro del entorno de trabajo o bien accesible desde la siguiente url - <https://marketplace.eclipse.org/> cuya funcionalidad es servir de repositorio centralizado para todos los plugins y herramientas construidos sobre Eclipse.

La web dispone de funcionalidad de búsqueda, y además presenta todo el contenido organizado en categorías, lo que facilita bastante el descubrir y explorar nuevos *plugins* o herramientas que no se conocen y pueden descargarse en nuestro entorno de trabajo.

Por último, destacar que se proporciona una amplia documentación agrupada por temáticas junto a un buscador avanzado accesible desde la siguiente url - <http://help.eclipse.org/luna/index.jsp>, convirtiéndolo en un entorno de desarrollo sencillo e intuitivo.

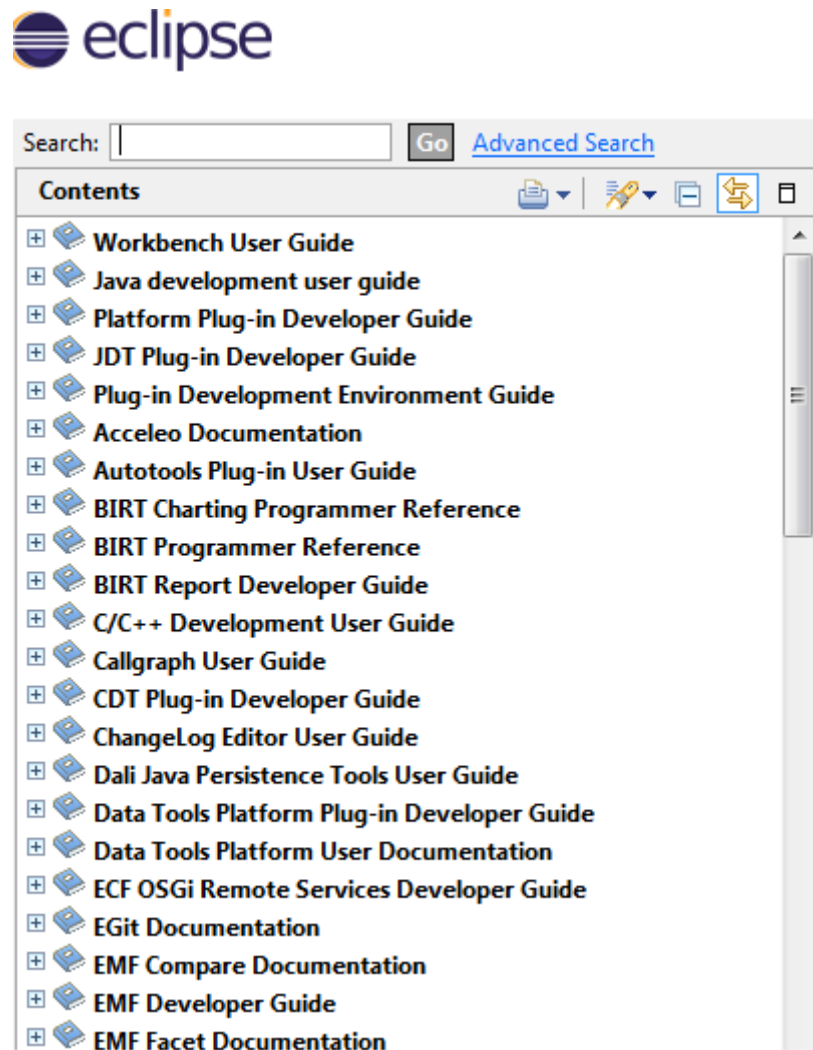


Figura 3.3.1 – Ayuda Eclipse.

Fuente: <http://help.eclipse.org/luna/index.jsp>

Una de las últimas actualización de Eclipse es la versión “eclipse-jee-luna-SR2”, accesible desde la url - <https://www.eclipse.org/>.



Figura 3.3.2 – Inicio Ejecución Local Eclipse Luna SR2

Integrado dentro del entorno de desarrollo Eclipse, se puede trabajar con: Spring Framework. Este Framework aporta un conjunto de clases que facilita sobremanera el trabajo cotidiano. Spring [REF 9] basándose en ficheros XML o anotaciones, es el encargado de construir todos los objetos que la aplicación va a utilizar.

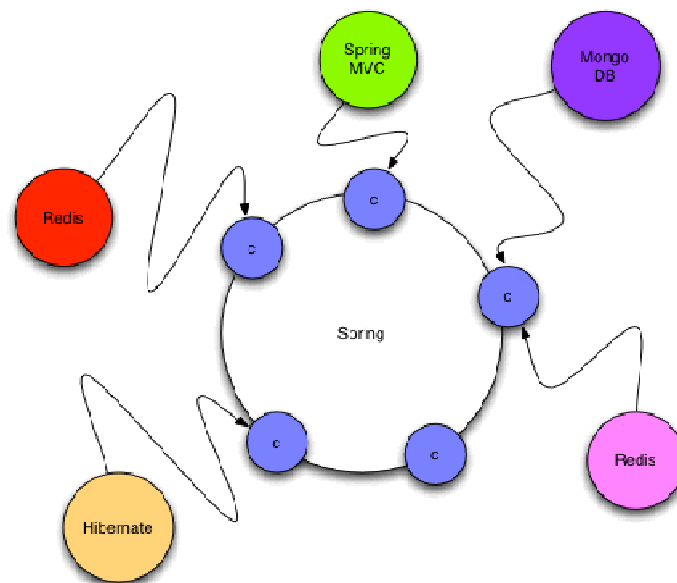


Figura 3.3.3 – Ejemplo. Spring y extensiones de objetos.

Fuente: <http://www.arquitecturajava.com/java-ee-7-vs-spring-framework/>

3.4. Oracle

Para garantizar la persistencia de la información, uno de los sistemas más utilizados es Oracle [REF 7].



Figura 3.4.1 – Logo Oracle. Fuente Oracle <http://www.oracle.com/es/index.html>.

Oracle Corporation es una de las mayores compañías de software del mundo. Hace tres décadas, Larry Ellison, vio una oportunidad que otras empresas no supieron apreciar al descubrir la descripción de un prototipo de trabajo para una base de datos relacional que ninguna empresa se había comprometido a comercializar. Ellison y sus cofundadores, Bob Miner y Ed Oates, se dieron cuenta del gran potencial económico que ofrecía el modelo de base de datos relacional pero no se dieron cuenta de que ellos cambiarían la informática empresarial para siempre.

Con la agilidad de una empresa mucho más pequeña, Oracle ha demostrado, gracias a su historial, que puede construir para el futuro sobre la base de años de innovación, el gran conocimiento de los éxitos y desafíos de sus clientes, y los mejores talentos en el área técnica y comercial alrededor del mundo. La empresa ha demostrado no solo su capacidad de aprovechar al máximo su gran tamaño y sus virtudes para servir a sus clientes, sino también su capacidad de tomar decisiones que eliminen las creencias convencionales y lleven sus productos y servicios hacia una nueva dirección.

Después de 30 años, Oracle sigue siendo el estándar de oro para las aplicaciones y la tecnología de base de datos de empresas de todo el mundo: La compañía es proveedora líder mundial de software para la administración de la información, y la segunda empresa de software independiente más grande del mundo. La tecnología de Oracle puede encontrarse en casi todos los sectores, y en los centros de datos de 98 de las 100 empresas Fortune (https://es.wikipedia.org/wiki/Fortune_500). Oracle es la primera empresa de software en desarrollar e implementar software empresarial 100% activado por internet en toda su línea de productos: base de datos, aplicaciones comerciales y herramientas para el soporte de decisiones y el desarrollo de aplicaciones.

Después de esta breve referencia histórica y las evidencias sobre el potencial que ofrece Oracle, es el motivo principal por el que muchas compañías elijen Oracle como BBDD y SGBD. Es importante indicar que existe una versión de Oracle, Database 11g, de descarga gratuita. Esta BBDD se presenta en diversas ediciones, pensada para satisfacer las necesidades comerciales y de tecnología de la información (IT) de todas las organizaciones. Oracle Database 11g Enterprise Edition también ofrece varias alternativas para empresas con requisitos comerciales y de IT especiales.

En este documento se resumen las características y opciones que integran cada edición de Oracle Database 11g.

- Ofrece innovación en facilidad de uso, capacidad y relación precio/rendimiento para aplicaciones de grupos de trabajo, departamentales y web en servidores únicos con un máximo de dos sockets.
- Se puede usar en un solo servidor o en varios servidores.
- Es posible gestionar la información de manera eficiente, confiable y segura en aplicaciones transaccionales indispensables, almacenes de datos con alto tráfico de consultas y cargas mixtas de trabajo.
- Automatiza las tareas de administración y ofrece las mejores funciones de seguridad y de cumplimiento de las normativas, por lo que consigue resultados óptimos.
- Ofrece un rendimiento y una escalabilidad excepcionales en servidores Windows, Linux y UNIX.

Para realizar la conexión será necesario utilizar un componente conocido como JDBC (Java Data Base Connector) el cual, debe proveer el creador del motor de la base de datos.

3.5. Apache Tomcat 7

Apache Tomcat [8], es un servidor web basado en el lenguaje Java que actúa como motor de *Servlets* y *JSPs* (*Java Server Pages*).

El servidor Tomcat ha sido desarrollado por "**Apache Software Foundation**" <http://apache.org/>, se trata de software libre gratuito y los usuarios pueden acceder a su código fuente.

Entre los principales beneficios de la utilización de Apache Tomcat:

- Multiplataforma
- Permite instalación local sin acceso a la red, permitiendo la depuración antes de la explotación de la aplicación en entorno real.
- Open Source – Código libre

Tanto el entorno Eclipse como el Tomcat vienen configurados para utilizar UTF-8 como *encoding* por defecto (formato de codificación de caracteres variable).



Figura 3.5.1 – Logo Apache-Tomcat. Fuente - <http://tomcat.apache.org/index.html>

3.6. GPS

El G.P.S. (Global Positioning System) se trata de un sistema de posicionamiento global que proporciona, a cualquier usuario que disponga de un sistema receptor de señal, la posición (longitud, latitud y altura), su velocidad y su hora correspondiente. Esta información es recibida de manera continua y en cualquier parte de mundo.

Fue desarrollado por el Departamento de Defensa de los EE.UU en los años 60 y está compuesto por tres segmentos:

- Espacial: compuesto por 24 satélites.
- Control: compuesto por 5 estaciones oficiales de seguimiento.
- Utilitario: compuesto por los instrumentos empleados por los usuarios.

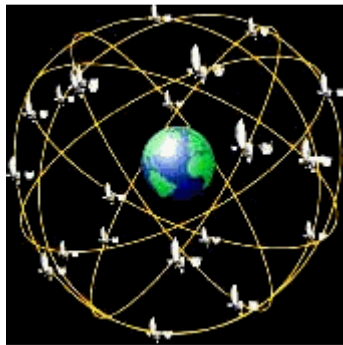


Figura 3.6.1 – Constelación de satélites GPS.

Fuente <http://www.proteccioncivil.org>

Los receptores GPS más sencillos están preparados para determinar con un margen mínimo de error la latitud, longitud y altura desde cualquier punto de la tierra donde nos encontremos situados. Otros dispositivos más completos muestran también el punto donde hemos estado, e incluso trazan de forma visual sobre un mapa la trayectoria seguida o la que vamos siguiendo en esos momentos. Esta es una capacidad que no poseían los dispositivos de posicionamiento anteriores a la existencia de los receptores GPS.

El funcionamiento del sistema GPS se basa también, al igual que los sistemas electrónicos antiguos de navegación, en el principio matemático de la triangulación. Por tanto, para calcular la posición de un punto será necesario que el receptor GPS determine con exactitud la distancia que lo separa de los satélites.

Como se explicó anteriormente, con la aplicación del principio matemático de la triangulación podemos conocer el punto o lugar donde nos encontramos situados, e incluso rastrear y ubicar el origen de una transmisión por ondas de radio. El sistema GPS utiliza el mismo principio, pero en lugar de emplear círculos o líneas rectas crea esferas virtuales o imaginarias para lograr el mismo objetivo. Desde el mismo momento

que el receptor GPS detecta una señal de radio frecuencia transmitida por un satélite desde su órbita, se genera una esfera virtual o imaginaria que envuelve al satélite. El propio satélite actuará como centro de la esfera cuya superficie se extenderá hasta el punto o lugar donde se encuentre situada la antena del receptor; por tanto, el radio de la esfera será igual a la distancia que separa al satélite del receptor. A partir de ese instante el receptor GPS medirá las distancias que lo separan como mínimo de dos satélites más. Para ello tendrá que calcular el tiempo que demora cada señal en viajar desde los satélites hasta el punto donde éste se encuentra situado y realizar los correspondientes cálculos matemáticos.

Principios de localización GPS.

- Triangulación. La base del GPS es la "triangulación" desde los satélites.
- Distancias. Para "triangular", el receptor de GPS mide distancias utilizando el tiempo de viaje de señales de radio.
- Tiempo. Para medir el tiempo de viaje de estas señales, el GPS necesita un control muy estricto del tiempo.
- Posición. Además de la distancia, el GPS necesita conocer exactamente donde se encuentran los satélites en el espacio. Órbitas de mucha altura y cuidadoso monitoreo, le permiten hacerlo.
- Corrección. Finalmente el GPS debe corregir cualquier demora en el tiempo de viaje de la señal que esta pueda sufrir mientras atraviesa la atmósfera.
- Fuentes de error. Para medir el momento a partir del cual el satélite emite la señal y el receptor GPS la recibe, es necesario que tanto el reloj del satélite como el del receptor estén perfectamente sincronizados.

Por lo tanto, para conocer una ubicación exacta, únicamente se necesita un receptor GPS que disponga de una antena de recepción y una calculadora que permita realizar los cálculos de distancia entre el receptor de señal y los diferentes satélites.

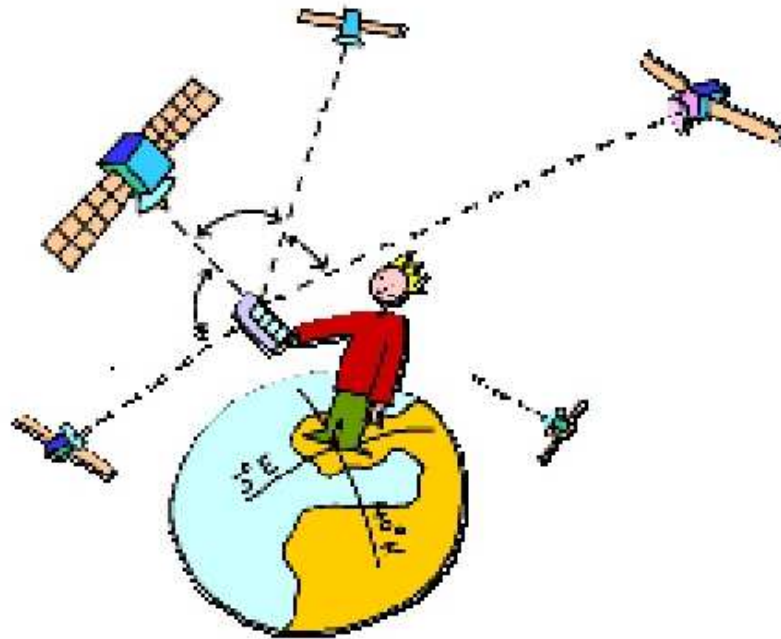


Figura 3.6.2 – Representación gráfica captura de señal GPS.

¿Es fiable la posición calculada? Con los avances alcanzados en la evolución del GPS, tenemos la posibilidad de conocer nuestra posición con un margen de error inferior a 20 metros o incluso menor, para ello es necesario la medida simultánea entre el receptor y al menos 3 o 4 satélites para que el resultado del posicionamiento sea válido y fiable.

Otros sistemas de posicionamiento, pero que todavía están desarrollándose.

- “Glonass”, sin entrar al detalle del funcionamiento de este sistema ya que no aplicaría al PFC, se trataría de la alternativa rusa al sistema desarrollado por EE.UU.
- “Galileo”, Galileo es un sistema global de navegación por satélite (GNSS) desarrollado por la Unión Europea, con el objeto de evitar la dependencia de los sistemas GPS y GLONASS. Al contrario de estos dos, será de uso civil.

3.7. Formula Haversine

La fórmula de Haversine, obtiene con gran exactitud la distancia entre dos puntos dados, identificados por su latitud y longitud. Esta fórmula es de gran utilidad en navegación marítima y será de gran utilidad también en la solución e implementación del PFC.

Conociendo los valores de Latitud y Longitud podremos calcular de una manera sencilla la distancia entre dos puntos.

- La longitud es la distancia que existe entre un punto cualquiera y el Meridiano de Greenwich, medida sobre el paralelo que pasa por dicho punto.

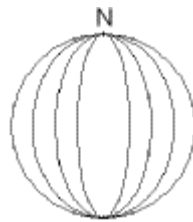


Figura 3.6.1 – Representación Longitud.

- La latitud es la distancia que existe entre un punto cualquiera y el Ecuador, medida sobre el meridiano que pasa por dicho punto.



Figura 3.6.1 – Representación Latitud.

Sin entrar en profundidad en detalles de términos matemáticos, la Fórmula del Haversine se define de la siguiente manera:

$$R = \text{radio de la Tierra}$$

$$\Delta\text{lat} = \text{lat2} - \text{lat1}$$

$$\Delta\text{long} = \text{long2} - \text{long1}$$

$$a = \sin^2(\Delta\text{lat}/2) + \cos(\text{lat1}) \cdot \cos(\text{lat2}) \cdot \sin^2(\Delta\text{long}/2)$$

$$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a})$$

$$d = R \cdot c$$

Únicamente faltaría un parámetro más, para utilizar la Fórmula del Haversine. Se necesita, además de las dos posiciones latitud y longitud, el radio de la Tierra. Este valor es relativo a la latitud, pues al no ser la Tierra perfectamente redonda, se podrían definir:

- Radio Ecuatorial cuyo valor es 6378 km.
- Radio Polar cuyo valor es de 6357 km.
- Radio Medio cuyo valor es de 6371 km.

Para realizar el cálculo, se trabaja con las coordenadas en la unidad de grados decimales.

Por ejemplo, si accedemos a un mapa y mostramos la localización de dos ciudades importante como Madrid y Barcelona se obtendrá la siguiente localización en Grados Decimales:

Puerta del Sol – Madrid:

- Latitud => 40.4169473
- Longitud => -3.70352849999999476

Ramblas – Barcelona:

- Latitud => 41.380654
- Longitud => 2.17384809999999865

Si aplicamos la fórmula de Haversine, se obtiene una distancia en línea recta de 505.92 Kms

3.8. API – Google Maps

Para la gestión de la rutas y que puedan ser visualizadas en un mapa, la opción más utilizada en la actualidad, es el API de Google Maps [REF 10]. Si se accede a la documentación del API en la Web, se detalla que se trata de un interface de programación de aplicaciones, “que te permiten superponer tus propios datos sobre un mapa de Google Maps personalizado. Puedes crear atractivas aplicaciones web y móviles con la potente plataforma de mapas de Google, incluso con imágenes de satélite, Street View, perfiles de elevación, indicaciones sobre cómo llegar, mapas con estilos, demografía, análisis y una amplia base de datos de ubicaciones. Con la cobertura global más precisa del mundo y una comunidad de mapas activa que incorpora actualizaciones diarias, los usuarios se beneficiarán de un servicio que mejora constantemente.”

Destacar como principales ventajas:

- Familiaridad y sencillez.
- Aporta soluciones precisas e innovadoras.
- Compatibilidad con dispositivos móviles.

Aparentemente se trata de la mejor solución posible y la opción más utilizada por aplicaciones y sistemas *Web*; no obstante, se debe tener en cuenta que también tiene un inconveniente o limitación:

- El API de Google Maps tiene un límite de 25.000 mapas básicos o 2.500 mapas estilizados por día. Si se supera este límite, la visualización de mapas dejaría de ser gratuito, por lo que se debería abonar una cantidad para que nuestra aplicación pueda seguir funcionando.

Teniendo en cuenta las principales ventajas y la limitación anteriormente mencionada, sigue siendo la opción más utilizada.

Otras alternativas de carácter completamente gratuito y sin limitaciones puede ser, Open Street Map [REF 11].

Capítulo 4. Análisis

El siguiente apartado hace referencia a la fase de análisis del PFC. En esta actividad se deben establecer los requisitos del proyecto que nos ocupa, teniendo como objetivo determinar el dominio de la aplicación, servicios y funcionalidades que debe proporcionar al usuario final, así como posibles restricciones del sistema. La fase de análisis se basará en los siguientes puntos:

- Requisitos no funcionales.
- Requisitos funcionales.
- Sistema: Conceptos y relaciones.
- Definición de escenarios, casos de uso y roles.

Un análisis exhaustivo y detallado es importante para tener un control de lo que se espera de una aplicación, evitándose confusiones sobre la entrega del software y cumpliéndose las expectativas del cliente o usuario final. Las especificaciones funcionales deben definirse de forma clara y concisa; además, es importante que el Usuario y el Equipo de desarrollo trabajen juntos para alcanzar el objetivo en el cierre de requisitos y dirigir el proyecto hacia un objetivo común.

Cada uno de los requisitos deberá ser validado, tanto por los equipos de desarrollo y responsables del proyecto como por el Usuario final.

Resumiendo, la fase de análisis se basará en los siguiente principios:

- Requisitos y compromisos claros.
- Equipos auto-gestionados e inter-relacionados.
- Entrega temprana y continua. Iteraciones cortas y aceptación del usuario.



Figura 4.1 – Diagrama aprobación de requisitos

4.1. Definición de requisitos

El principal objetivo en la definición de requisitos es establecer lo que el sistema debe hacer, sus propiedades deseadas y esenciales y las restricciones en el funcionamiento del sistema. Por lo tanto, la definición de requisitos se considera como el proceso de comunicación entre los clientes/usuarios y los desarrolladores y responsables de la aplicación para definir el alcance de la aplicación o sistema.

En la fase de definición de requisitos se debería tener en cuenta:

- Fuentes de información o soportes documentales de apoyo.
- Entrevistas con Usuarios.
- Inquietudes de usuarios y conflictos entre diferentes interlocutores.
- Especificación de sistemas similares.
- Viabilidad de los requisitos.
- Independencia de la tecnología a utilizar.

Para mejorar el entendimiento y poder distribuir el trabajo entre los diferentes equipos de desarrollo, los requisitos de Sistemas de software se podrían clasificar en los siguientes tipos:

- Funcionales.
- No Funcionales.
- Dominio del Sistema.

Siguiendo esta clasificación, en el PFC que nos aplica, utilizaremos esta división para establecer el alcance y funcionalidad del Sistema a desarrollar.

Los requisitos de dominio , son requisitos que provienen del dominio del sistema y reflejan características y restricciones de éste, pudiendo ser funcionales como no funcionales; por lo que en el caso que nos aplica no se identificarán requisitos de dominio de sistema y estarán contenidos dentro de las dos categorías Funcionales y No funcionales.

Tanto los requisitos funcionales como los requisitos no funcionales se recogerán con la siguiente plantilla (Figura 4.1.1), que permitirá identificarlos, tener un control y realizar un seguimiento sobre ellos.

Identificador		Categoría		Fecha	
Requisito		Prioridad			
Descripción					
GAP Origen		Dependencia de otros requisitos			
Versión				Fecha	
Descripción					

Figura 4.1.1 – Plantilla requisitos

El seguimiento de los requisitos permitirá anticipar desviaciones o incumplimientos en las funcionalidades definidas y depurar responsabilidades.

En un principio la división de los requisitos anteriormente establecida, parecería una clasificación y organización “cerrada”, pero se tiene que tener en cuenta que un requisito que en un principio encajaría en el tipo de requisitos ‘No Funcionales’, como pudiera ser el caso de la seguridad, a medida que se está desarrollando podría generar otros con un enfoque y tipificación claramente funcional, como la necesidad de autenticación y validaciones de usuario. Por lo tanto, el proceso de definición de requisitos debe ser un proceso abierto y flexible, siendo de vital importancia tener un control de versiones sobre los documentos generados. El control de versiones del Documento Definición Requisitos (DDR) permitirá hacer un seguimiento de cada uno de los requisitos y la evolución y versiones de cada uno de ellos.

Para el control de versiones del PFC se seguirá el siguiente formato – Figura 4.1.2.

Control de versiones

Versión	Fecha	Modificación	Autor
v1.0	29/06/15	Primera versión del documento	Alberto Herranz

Figura 4.1.2 – Plantilla requisitos – Control de versiones.

Por último, de la plantilla de requisitos de la Figura 4.1.1, se destaca el campo prioridad, este campo permite identificar los requisitos por la prioridad marcada por el usuario, esta prioridad posibilita realizar entregas parciales o negociar con el cliente o usuario final las entregas de software a realizar en casos de desviaciones negativas en la planificación de los proyectos o conflictos con otros requisitos generados cuando hay muchos actores o interlocutores en la fase de análisis.

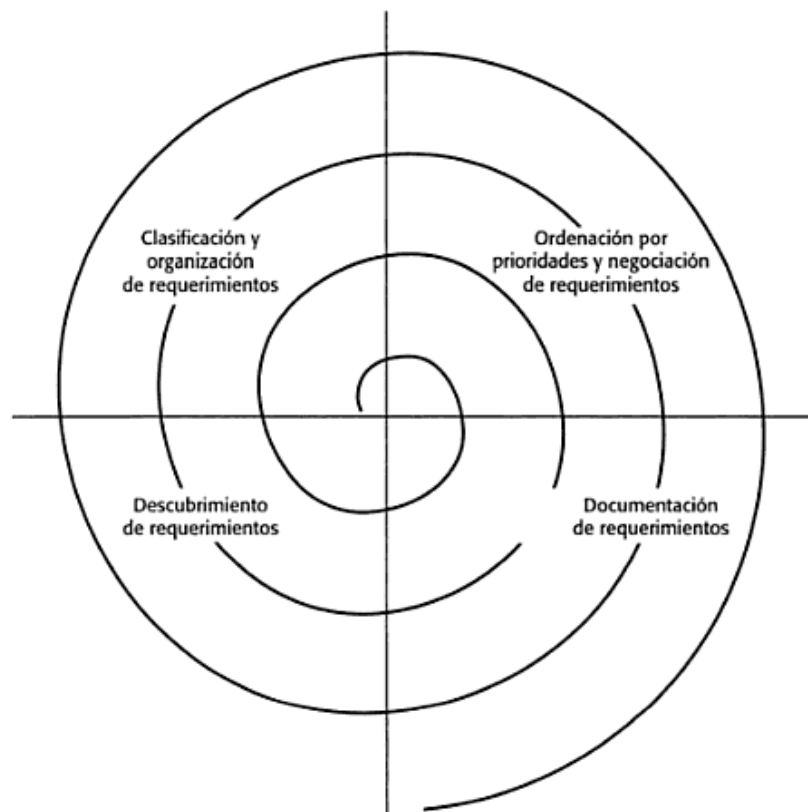


Figura 4.1.2 – Representación gráfica. Proceso de obtención y análisis de requisitos.

En la figura 4.1.2 se muestra un modelo general del proceso de obtención y análisis en el que se puede ver las diferentes actividades dentro de una espiral, de forma que cada una de ellas se entrelazan a medida que el proceso avanza desde el anillo interior al anillo exterior del espiral. Los procesos de definición deben ser flexibles ante cambios e imprevistos.

4.1.1. Definición de requisitos funcionales

Los requisitos funcionales de un sistema describen de una forma clara lo que el sistema debe hacer, de ahí que sea de gran importancia intentar dividir los requisitos y evitar redactar en un mismo requisito varias funcionalidades.

“Divide y vencerás”

La definición de estos requisitos depende en gran medida del tipo de software a desarrollar y de un posible enfoque organizativo, pudiéndose redactar de una forma completamente abstracta, como los requisitos de usuario, o bien dar un detalle funcional ampliado de lo que se espera de la operación, incluyéndose entradas y salidas de las diferentes funcionalidades.

En el PFC que nos aplica se establecen los requisitos funcionales de forma detallada y atómica, evitándose malinterpretaciones con definiciones abstractas por parte de usuario. Una definición detallada posibilitará:

- Mezcla de funcionalidades en un mismo requisito.
- Evitar ambigüedades o simplificación de funcionalidad por parte de los desarrolladores.
- Reducción de los cambios a realizar y minimizar el control de versiones.
- Reducción de costes de proyecto.
- Evitar retrasos en las entregas.

No obstante, como se detalló en el capítulo 4, el proceso de definición y recogida de requisitos debe ser abierto y flexible, por lo que cualquier modificación sobre una versión inicial de requisito no debe ser sinónimo de un problema sino ser considerada como enriquecedora y aportar valor sobre el desarrollo final del software con un documento de definición, DDR, de calidad, de ahí la importancia de la comunicación entre los diferentes interlocutores y la fusión de los puntos de vista de cada uno de ellos.

Se pasa a continuación a definir los requisitos funcionales del PFC ‘Non-Stop’. Cada requisito se identifica con el prefijo “FU_NS_XX”, Funcional_Non-Stop, siendo XX un secuencial para identificar cada requisito funcional del Sistema.

Requisito - Funcionalidad Alta de usuario.

Identificador	FU_NS_01	Categoría	Funcional	Fecha	22-07-2015
Requisito	Alta de Usuario	Prioridad		Alta	
Descripción	<p>El sistema de monitorización ‘Non-Stop’ permitirá el alta de nuevos usuarios para el acceso y uso de la aplicación. El sistema proveerá de un formulario de alta en el que se detallará la siguiente información que debe ser completada por el usuario.</p> <ul style="list-style-type: none">▪ Usuario.▪ Contraseña.▪ Fecha de nacimiento.▪ Ciudad.▪ Correo electrónico.▪ Teléfono de emergencia. <p>El sistema realizará las validaciones sobre la información aportada por el usuario y proporcionará un mensaje informativo con la confirmación de alta en el sistema.</p> <p>El sistema de monitorización ofrecerá a los usuarios dados de alta las diferentes funcionalidades incluidas en el sistema y realizará su correspondiente alta en BBDD para accesos futuros.</p> <ul style="list-style-type: none">▪ Iniciar nueva ruta.▪ Consultar rutas.▪ Consultar alarmas generadas.				
Versión	V01R00	Dependencia de otros requisitos			

Requisito - Funcionalidad validar alta de usuario.

Identificador	FU_NS_02	Categoría	Funcional	Fecha	22-07-2015
Requisito	Validar alta usuario.	Prioridad		Alta	
Descripción	<p>El sistema de monitorización ‘Non-Stop’ realizará la validación correspondiente al alta de nuevos usuarios.</p> <ul style="list-style-type: none">▪ El sistema validará que el usuario no esté registrado con anterioridad. Si el usuario estuviera registrado, el sistema deberá informar al usuario indicándole un mensaje informativo con el título usuario registrado.▪ El sistema validará el formato de los campos introducidos.▪ El sistema validará la completitud de todos los campos obligatorios, El sistema identificará todos los campos obligatorios con el símbolo (*)▪ El sistema no realizará validaciones de coherencia de datos introducidos. <p>El sistema informará al usuario sobre la obligatoriedad de los datos en el caso de no haberse informado y si algún campo del formulario de alta ha sido completado de forma errónea.</p>				
Versión	V01R00	Dependencia de otros requisitos		FU_NS_01	

Requisito - Funcionalidad validar formato.

Identificador	FU_NS_03	Categoría	Funcional	Fecha	27-08-2015
Requisito	Validar formato	Prioridad		Alta	
Descripción	<p>El sistema de monitorización ‘Non-Stop’ realizará las validaciones de formato correspondiente al alta de nuevos usuarios.</p> <ul style="list-style-type: none">▪ El sistema validará el campo usuario, permitiéndose un máximo de 20 caracteres alfanuméricos y un mínimo de 3.▪ El sistema validará el campo contraseña, permitiéndose un máximo de 12 caracteres alfanuméricos y un mínimo de 6 que garantice una clave de acceso segura.▪ El sistema validará el campo correo electrónico, este campo debe incluir el símbolo ‘@’ y no contener espacios entre el identificador de correo y el dominio (Ej. micorreo@dominio.es).▪ El sistema validará el campo fecha de nacimiento con el siguiente formato dd/mm/aaaa.▪ El sistema validará el campo teléfono de emergencia, este campo deberá tener el siguiente formato correspondiente a teléfonos móviles XXX-XXXXXX. El sistema no validará teléfonos internacionales. <p>El sistema informará al usuario del error producido en el campo correspondiente posibilitándose la modificación de los datos indicados como erróneos.</p>				
Versión	V01R00	Dependencia de otros requisitos		NF_NS_06 FU_NS_01	

Requisito - Funcionalidad acceder a la aplicación.

Identificador	FU_NS_04	Categoría	Funcional	Fecha	27-08-2015
Requisito	Acceder aplicación.	Prioridad		Alta	
Descripción	<p>El sistema de monitorización ‘Non-Stop’ proporcionará un formulario de acceso a la aplicación para el usuario con la siguiente información que deberá ser completada de forma correcta.</p> <ul style="list-style-type: none">▪ Usuario▪ Contraseña. <p>El sistema realizará las validaciones de usuario registrado informando al usuario con un mensaje de advertencia de usuario o contraseña incorrectos.</p> <p>El sistema no se realizará bloqueos de usuario por contraseñas e intentos fallidos de acceso.</p> <p>El sistema de monitorización ofrecerá a los usuarios que accedan correctamente a la aplicación las diferentes funcionalidades incluidas en el sistema.</p> <ul style="list-style-type: none">▪ Iniciar nueva ruta.▪ Consultar rutas.▪ Consultar alarmas generadas.				
Versión	V01R00	Dependencia de otros requisitos		NF_NS_06 FU_NS_01	

Requisito - Funcionalidad iniciar sistema de alarmas.

Identificador	FU_NS_05	Categoría	Funcional	Fecha	27-08-2015
Requisito	Iniciar sistema alarmas y envío de SMS.	Prioridad		Alta	
Descripción	<p>El sistema de monitorización 'Non-Stop' proporcionará al usuario la modalidad, sistema de alarmas y monitorización de rutas.</p> <ul style="list-style-type: none">▪ El sistema al iniciar la aplicación en la modalidad Alarma, registrará el Track en BBDD.▪ El sistema posibilitará acceder al track de acceso público y pintará la ruta en un mapa cuando sea solicitado.▪ El sistema generará una alarma al analizar el track recibido y detectar una parada inesperada. El sistema registrará la alarma en BBDD y realizará el envío de SMS al teléfono de emergencia registrado en el alta de usuario.▪ El sistema podrá generar un número variable de alarmas durante la realización de la ruta. <p>El sistema de monitorización 'Non-Stop' requerirá que el dispositivo desde donde se comience la actividad tenga integrado un dispositivo GPS y datos móviles habilitados.</p>				
Versión	V01R00	Dependencia de otros requisitos		FU_NS_04 FU_NS_06 NF_NS_07	

Requisito - Funcionalidad iniciar sistema modo privado.

Identificador	FU_NS_06	Categoría	Funcional	Fecha	27-08-2015
Requisito	Iniciar sistema modo privado.	Prioridad		Alta	
Descripción	<p>El sistema de monitorización ‘Non-Stop’ proporcionará al usuario la modalidad, sistema de monitorización de rutas en modo privado.</p> <ul style="list-style-type: none">▪ El sistema al iniciar la aplicación en la modalidad privada, registrará el Track en BBDD. El track no será accesible por terceras personas y será de carácter privado.▪ El sistema únicamente permitirá el acceso a la ruta realizada por parte del usuario que generó la ruta.▪ El sistema no analizará la información del track recibido y no generará alarmas y envío de SMS, únicamente guardará en BBDD la información del track. <p>El sistema de monitorización ‘Non-Stop’ requerirá que el dispositivo desde donde se comience la actividad tenga integrado un dispositivo GPS y datos móviles habilitados.</p>				
Versión	V01R00	Dependencia de otros requisitos		FU_NS_04 NF_NS_07	

Requisito - Funcionalidad consultar alarmas.

Identificador	FU_NS_07	Categoría	Funcional	Fecha	27-08-2015
Requisito	Consultar alarmas.	Prioridad		Alta	
Descripción	<p>El sistema de monitorización ‘Non-Stop’ proveerá de un formulario de búsqueda de alarmas generadas.</p> <p>El sistema de búsqueda de alarmas tendrá los siguientes parámetros de entrada.</p> <ul style="list-style-type: none">▪ Fecha.▪ Usuario ruta. <p>El sistema de búsqueda de alarmas obtendrá de salida una lista de alarmas ordenada con los siguientes parámetros de salida.</p> <ul style="list-style-type: none">▪ Mapa ruta realizada.▪ Ubicación actual.▪ Ubicación alarma. <p>El sistema realizará la ordenación de la lista de salida por fecha de generación de la alarma. Asociado a cada alarma de la lista, el sistema proveerá de una opción para acceder de forma unitaria al detalle de cada alarma.</p>				
Versión	V01R00	Dependencia de otros requisitos		FU_NS_04 FU_NS_05 FU_NS_09	

Requisito - Funcionalidad consultar rutas.

Identificador	FU_NS_08	Categoría	Funcional	Fecha	27-08-2015
Requisito	Consultar rutas.	Prioridad		Alta	
Descripción	<p>El sistema de monitorización ‘Non-Stop’ proveerá de un formulario de búsqueda de rutas generadas.</p> <p>El sistema de búsqueda de rutas tendrá los siguientes parámetros de entrada.</p> <ul style="list-style-type: none">▪ Fecha.▪ Usuario ruta. <p>El sistema de búsqueda de rutas obtendrá de salida una lista rutas ordenada con los siguientes parámetros de salida.</p> <ul style="list-style-type: none">▪ Mapa ruta realizada.▪ Alarmas generadas. <p>El sistema realizará la ordenación de la lista de salida por fecha de generación de la ruta. Asociado a cada ruta de la lista, el sistema proveerá de una opción para acceder de forma unitaria al detalle de cada ruta..</p>				
Versión	V01R00	Dependencia de otros requisitos		FU_NS_04 FU_NS_06 FU_NS_10	

Requisito - Funcionalidad detalle de alarmas.

Identificador	FU_NS_09	Categoría	Funcional	Fecha	27-08-2015
Requisito	Detalle de alarmas.	Prioridad		Alta	
Descripción	<p>El sistema de monitorización ‘Non-Stop’ proveerá de un formulario con el detalle de cada alarma seleccionada.</p> <p>El sistema recuperará la siguiente información de salida para cada alarma seleccionada.</p> <ul style="list-style-type: none">▪ Mapa ruta realizada.▪ Ubicación actual.▪ Ubicación alarma.				
Versión	V01R00	Dependencia de otros requisitos		FU_NS_04 FU_NS_07	

Requisito - Funcionalidad detalle de rutas.

Identificador	FU_NS_10	Categoría	Funcional	Fecha	27-08-2015
Requisito	Detalle de rutas.	Prioridad		Alta	
Descripción	<p>El sistema de monitorización ‘Non-Stop’ proveerá de un formulario con el detalle de cada ruta seleccionada.</p> <p>El sistema recuperará la siguiente información de salida para cada alarma seleccionada.</p> <ul style="list-style-type: none">▪ Mapa ruta realizada.				
Versión	V01R00	Dependencia de otros requisitos		FU_NS_04 FU_NS_08	

Requisito - Funcionalidad borrar alarmas.

Identificador	FU_NS_11	Categoría	Funcional	Fecha	27-08-2015
Requisito	Borrar alarmas.	Prioridad		Alta	
Descripción	<p>El sistema de monitorización 'Non-Stop' proveerá de un formulario para realizar el borrado de una alarma seleccionada.</p> <p>El sistema permitirá únicamente el borrado de las alarma al usuario propietario de ella.</p> <p>El sistema informará al usuario con un mensaje informativo con título, alarma seleccionada borrada correctamente.</p>				
Versión	V01R00	Dependencia de otros requisitos		FU_NS_04 FU_NS_07	

Requisito - Funcionalidad borrar rutas.

Identificador	FU_NS_12	Categoría	Funcional	Fecha	27-08-2015
Requisito	Borrar rutas.	Prioridad		Alta	
Descripción	<p>El sistema de monitorización 'Non-Stop' proveerá de un formulario para realizar el borrado de una ruta seleccionada.</p> <p>El sistema permitirá únicamente el borrado de la ruta al usuario propietario de ella.</p> <p>El sistema informará al usuario con un mensaje informativo con título, ruta seleccionada borrada correctamente.</p>				
Versión	V01R00	Dependencia de otros requisitos		FU_NS_04 FU_NS_08	

4.1.2. Definición de requisitos no funcionales

Los requisitos no funcionales son restricciones de los servicios o funcionalidades ofrecidas por el Sistema. Principalmente se aplican al sistema en su totalidad evitándose la restricción o limitación de un servicio o funcionalidad a nivel individual.

Se podrían definir como las propiedades emergentes del sistema, tiempo de respuesta, la capacidad de almacenamiento o la disponibilidad del sistema. Suelen considerarse como requisitos con un carácter más crítico, dado que el incumplimiento de alguno de ellos podría significar que el sistema entero fuera inutilizable, mientras que en el caso de los requisitos funcionales se podría trabajar alrededor de una función del sistema que no cumple sus expectativas al 100% pero permite conseguir un objetivo final similar o camino alternativo.

A continuación se definen los requisitos no funcionales del PFC – ‘Non-Stop’. Cada requisito se identifica con el prefijo “NF_NS_XX”, No_Funcional_Non-Stop, siendo XX un secuencial para identificar cada requisito no funcional del Sistema.

Requisito – No Funcional capacidad de tamaño ruta.

Identificador	NF_NS_01	Categoría	Sistema	Fecha	22-07-2015
Requisito	Capacidad de tamaño ruta.	Prioridad		Alta	
Descripción	El sistema de monitorización ‘Non-Stop’ creará una ruta al activar el sistema de alarmas sin restricción de tamaño máximo.				
Versión	V01R00	Dependencia de otros requisitos			

Requisito – No Funcional disponibilidad temporal de la aplicación.

Identificador	NF_NS_02	Categoría	Sistema	Fecha	22-07-2015
Requisito	Disponibilidad temporal.	Prioridad		Alta	
Descripción	<p>El sistema de monitorización ‘Non-Stop’ tendrá una disponibilidad temporal superior al 90% en un periodo de 24 horas.</p> <p>El sistema generará restricciones temporales inferiores al 10% en un periodo de 24 horas por tareas de mantenimiento .</p>				
Versión	V01R00	Dependencia de otros requisitos			

Requisito – No Funcional disponibilidad temporal de la aplicación.

Identificador	NF_NS_03	Categoría	Sistema	Fecha	22-07-2015
Requisito	Restricción acceso.	Prioridad		Alta	
Descripción	El sistema de monitorización ‘Non-Stop’ proveerá de un método de registro de usuarios y realizará la validación e identificación de usuarios registrados.				
Versión	V01R00	Dependencia de otros requisitos		FU_NS_01	

Requisito – No Funcional persistencia de datos.

Identificador	NF_NS_04	Categoría	Sistema	Fecha	22-07-2015
Requisito	Persistencia de datos.	Prioridad		Alta	
Descripción	<p>El sistema de monitorización ‘Non-Stop’ proveerá de un método de registro de información, rutas y alarmas, en BBDD.</p> <p>El sistema de monitorización mantendrá la información de rutas y alarmas durante el periodo de tiempo que considere el usuario.</p>				
Versión	V01R00	Dependencia de otros requisitos			

Requisito – No Funcional seguridad de usuarios.

Identificador	NF_NS_05	Categoría	Sistema	Fecha	22-07-2015
Requisito	Seguridad usuarios	Prioridad		Alta	
Descripción	<p>El sistema de monitorización ‘Non-Stop’ proveerá de un método de registro de información, rutas y alarmas, en BBDD.</p> <p>La información de los usuarios no debe ser pública, excepto aquellos casos que se solicite para que puedan ser monitorizados.</p>				
Versión	V01R00	Dependencia de otros requisitos			

Requisito – No Funcional accesibilidad de aplicación.

Identificador	NF_NS_06	Categoría	Sistema	Fecha	22-07-2015
Requisito	Acceso aplicación.	Prioridad		Alta	
Descripción	El sistema de monitorización ‘Non-Stop’ será accesible desde cualquier dispositivo, móvil, PC, etc.				
Versión	V01R00	Dependencia de otros requisitos			

Requisito – No Funcional restricción comunicación GPS.

Identificador	NF_NS_07	Categoría	Sistema	Fecha	22-07-2015
Requisito	Comunicación GPS	Prioridad		Alta	
Descripción	El sistema de monitorización ‘Non-Stop’ será funcional en la generación de alarmas desde dispositivos móviles con señal GPS y tarifa de datos móviles para envío de SMS.				
Versión	V01R00	Dependencia de otros requisitos			

4.2. Escenarios y casos de uso

En este apartado se muestran los casos de uso que describen en detalle el funcionamiento del sistema.

Un caso de uso es una representación de las acciones de un sistema desde el punto de vista de usuario. Se representa la interacción con el sistema a desarrollar donde se muestran los requisitos funcionales. Cada caso de uso que representa el sistema, es una colección de situaciones y cada una de estas una secuencia de pasos. A las entidades que inician las secuencias se las conoce como actores.

A la hora de definir los casos de uso, es importante no confundirlos con los requisitos de la aplicación descritos en los apartados anteriores (un mismo caso de uso puede contemplar varios requisitos, y un mismo requisito puede estar contemplado por varios casos de uso).

Por lo tanto, los requisitos funcionales del sistema quedarán reflejados en uno o varios casos de uso y su correspondientes diagramas. La finalidad de los casos de uso es representar lo que el sistema es capaz de hacer y mejorar el entendimiento de lo que se espera de la aplicación y posibilitando a otras personas que lo comprendan, independientemente de la tecnología que se utilice.

Resumiendo, la representación de casos de uso permitirá tener a diferentes personas la misma visión sobre las funcionalidades y el resultado esperado del sistema. Los desarrolladores deben tener la misma idea mientras se desarrolla el sistema evitando confusiones posteriores entre analistas, desarrolladores y usuario final.



Figura 4.2.1 – Representación gráfica caso de uso simple.

4.2.1. Actores

En el PFC que nos aplica, se diferenciarán los siguientes actores que se agrupan dentro de dos tipologías, interno y externo. También existe la posibilidad de creación de grupos de usuario como un actor principal del sistema, que aunque en esta fase inicial del PFC no aplica, si se detallará a continuación como un actor del sistema.

- **Actor externo activo.** Se refiere al usuario de la aplicación que interactúa con el sistema de forma activa o en movimiento. Es el encargado de generar la señal GPS desde un dispositivo móvil. Su participación es necesaria para la funcionalidad del sistema.



Figura 4.2.1.1 – Representación gráfica actor externo activo.

- **Actor externo pasivo.** Se refiere al usuario de la aplicación que interactúa con la aplicación desde otro terminal diferente al actor externo activo, es el receptor pasivo de la aplicación, pudiendo acceder al sistema si lo considera, no siendo necesaria su intervención.



Figura 4.2.1.2 – Representación gráfica actor externo pasivo.

- **Actor interno o administrador del sistema.** Se refiere al administrador del sistema, encargado de resolver incidencias generadas por parte de los usuarios externos del sistema.

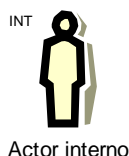


Figura 4.2.1.3 – Representación gráfica actor interno.

- **Grupo actores.** Se refiere al grupo de usuarios que pueden interactuar con el sistema de manera conjunta

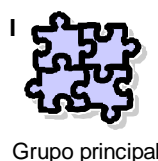
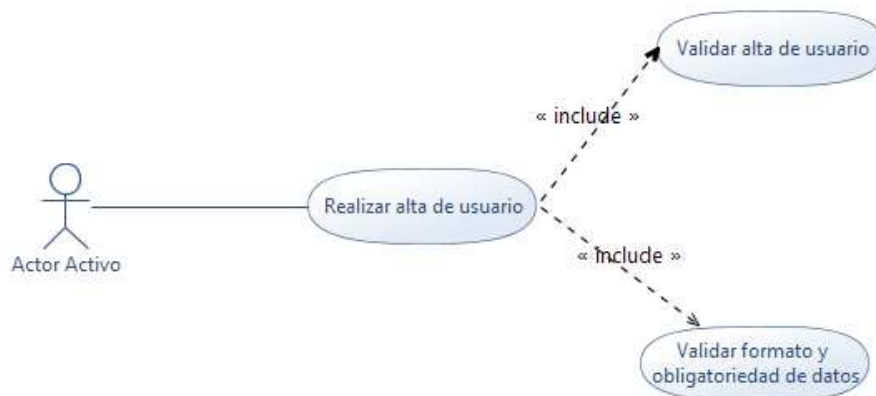


Figura 4.2.1.4 – Representación gráfica grupo actores.

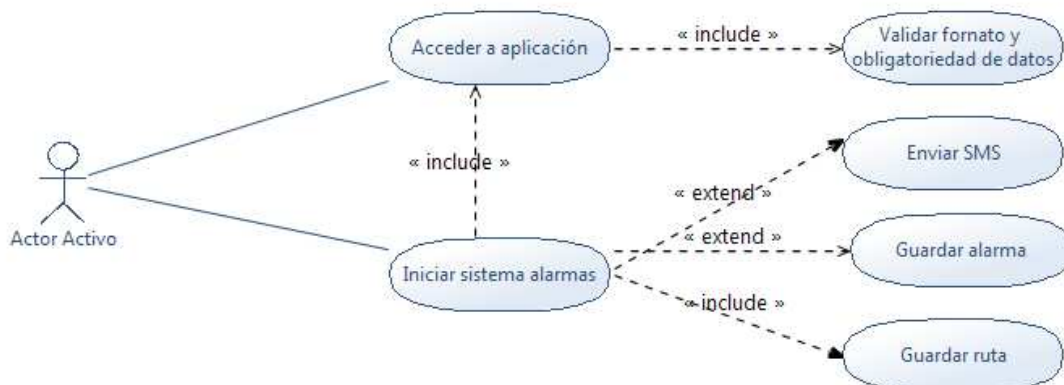
4.2.2. Representación gráfica casos de uso

En este apartado se definirán gráficamente los casos de uso y cada una de las partes que intervienen. En cada caso de uso, se visualizan los actores que inician la ejecución y cada parte implicada en el proceso que ejecuta o apoya en cada actividad.

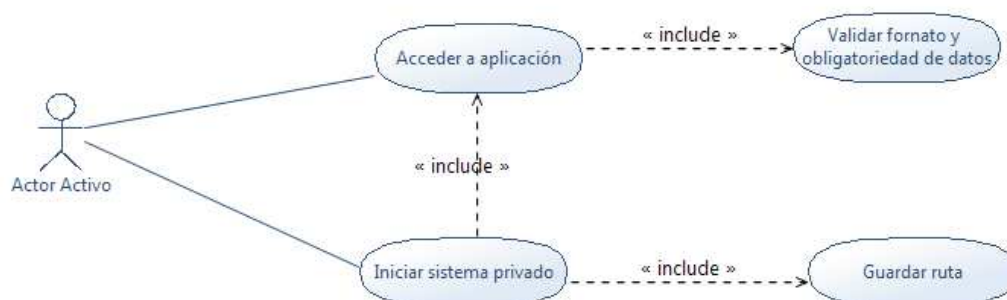
Caso de uso 1. Alta de usuario.



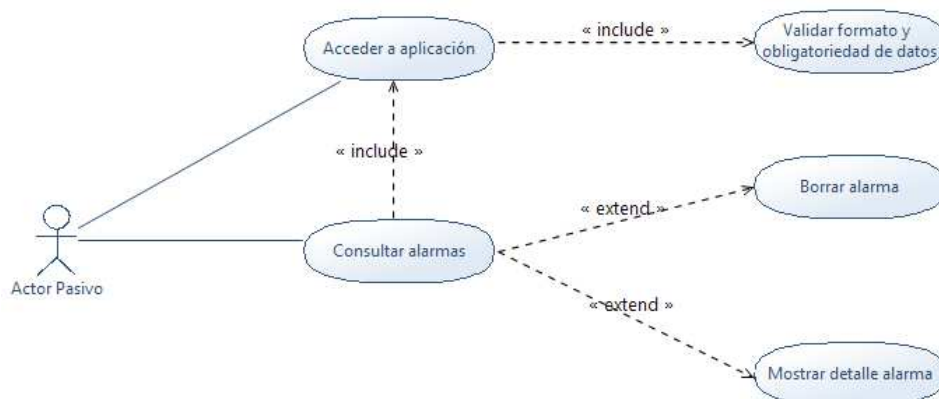
Caso de uso 2. Iniciar sistema modo alarmas.



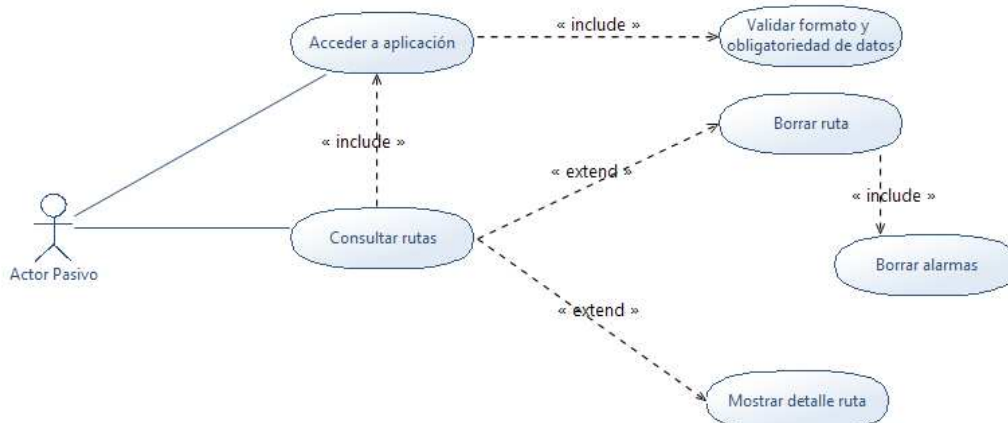
Caso de uso 3. Iniciar sistema modo privado.



Caso de uso 4. Consulta de alarmas.



Caso de uso 5. Consulta de rutas.



4.5. Matriz de trazabilidad casos de uso y requisitos funcionales

Cada caso de uso satisface uno o varios requisitos; por lo tanto, en este apartado se describe la matriz de trazabilidad entre ellos, permitiendo identificar de manera clara cada uno de los requisitos identificados por caso de uso.

Requisito\Caso de uso	Caso de uso 1 - Alta de usuario	Caso de uso 2 - Iniciar sistema alarmas	Caso de uso 3 - Iniciar sistema modo privado	Caso de uso 4 - Consulta de alarmas y opciones	Caso de uso 5 - Consulta de rutas y opciones
FU_NS_01 - Alta de usuario	X				
FU_NS_02 - Validar alta de usuario	X				
FU_NS_03 - Validar formato	X	X	X	X	X
FU_NS_04 - Acceder a aplicación		X	X	X	X
FU_NS_05 - Iniciar sistema de alarmas y envío de SMS		X			
FU_NS_06 - Iniciar sistema modo privado			X		
FU_NS_07 - Consultar alarmas				X	
FU_NS_08 - Consultar rutas					X
FU_NS_09 - Detalle de alarmas				X	
FU_NS_10 - Detalle de rutas					X
FU_NS_11 - Borrar alarmas				X	X
FU_NS_12 - Borrar rutas					X

Capítulo 5. Diseño

Una vez realizado el análisis y establecidos los requisitos funcionales y no funcionales se procederá a elaborar el diseño, la tecnología a utilizar y como se relacionan cada uno de los elementos que forman el sistema.

El diseño recogerá las especificaciones técnicas necesarias para que se pueda desarrollar e implementar adecuadamente el sistema objetivo del proyecto además de conocer cómo se estructurará la información en el modelo.

El documento de diseño técnico debería incluir la siguiente información.

Objetivo

El objetivo principal es desarrollar un sistema de comunicación de señal GPS entre un dispositivo móvil capaz de recibir una señal GPS y realizar el envío de esta de forma periódica a un sistema “servidor” de forma periódica.

Arquitectura del sistema

- Arquitectura en capas del sistema. Arquitectura Eclipse Java.
- Diagramas de paquetes.
- Diagramas de clases, tanto a nivel global de aplicación como a nivel de componentes específicos.
- Descripción técnica de cada uno de los componentes que forman el sistema.
- Diagramas de entidad-relación para el modelado de datos del sistema de información y esquemas de base de datos; si aplica.

5.1. Arquitectura del sistema

Se establece como arquitectura del sistema un Modelo Vista Controlador (MVC). MVC es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo.

- **Modelo:** contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia.

El modelo es el responsable de:

- Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo sea independiente del sistema de almacenamiento.
- Define las reglas de negocio: recogidas en la definición de cada uno de los FU del DDR.
- Lleva un registro de las vistas y controladores del sistema. Si estamos ante un modelo activo, notificará a las vistas los cambios que en los datos pueda producir un agente externo (por ejemplo, un fichero por lotes que actualiza los datos, un temporizador que desencadena una inserción, etc.).
- **Vista o interfaz de usuario:** compone la información que se envía al cliente y los mecanismos de interacción con éste.

Las vistas son responsables de:

- Recibir datos del modelo y los muestra al usuario.
- Tienen un registro de su controlador asociado (normalmente porque además lo instancia).
- Pueden dar el servicio de "Actualización", para que sea invocado por el controlador o por el modelo (cuando es un modelo activo que informa de los cambios en los datos producidos por otros agentes).
- **Controlador:** actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

El controlador es responsable de:

- Recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- Contiene reglas de gestión de eventos, del tipo "SI Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas.

El flujo que sigue el control generalmente es el siguiente:

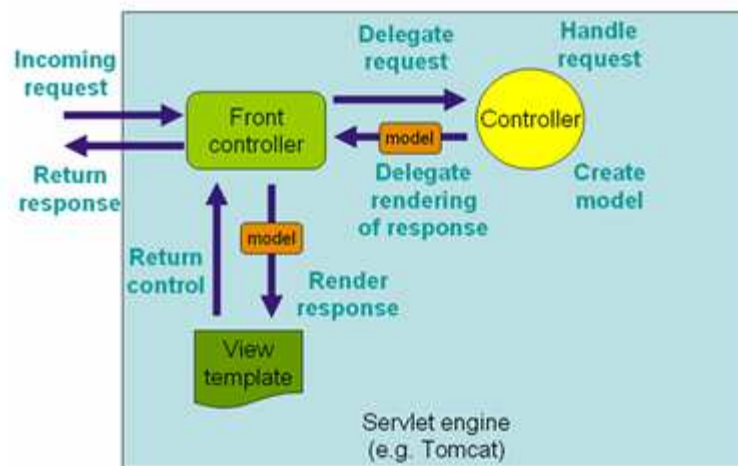


Figura 5.1.1 – Representación gráfica flujo MVC.
Fuente - <http://projects.spring.io/spring-framework/>

- El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.)
- El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega.
- El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario
- El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario.
- La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo. El modelo no debe tener conocimiento directo sobre la vista.
- La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

5.2. Estructura lógica del modelo

Se precisa de un modelo sencillo que permita almacenar la información requerida por el sistema para las siguientes funcionalidades analizadas.

- Acceso/Registro usuarios.
- Generar/Recuperar Rutas.
- Generar/Recuperar Alarmas.

La información estará agrupada en las siguientes estructuras de información y no estará vinculada a ningún SGBD (Sistema Gestor de BBDD).

- Usuario => Se requiere almacenar la información relativa al usuario, con todos los datos que se consideren, como pudiera ser nombre usuario, fecha nacimiento, etc.
- Ruta => Se requiere almacenar las rutas realizadas por el usuario y que estas puedan ser recuperadas a través de criterios de búsqueda simple.
- Alarma => Se requiere almacenar la información relativa a la alarma generada por el sistema. De la misma manera que las rutas, las alarmas deben ser accesibles mediante criterios de búsquedas sencillos.

Como se puede ver (Figura 5.2.1), la información lógica que se requiere para dar soporte a los diferentes requisitos funcionales es sencilla e intuitiva. En un principio puede parecer muy simple, pero aporta el potencial suficiente para dar soporte a las diferentes funcionalidades solicitadas; además, la sencillez del modelo facilitaría notablemente el desarrollo de la aplicación.

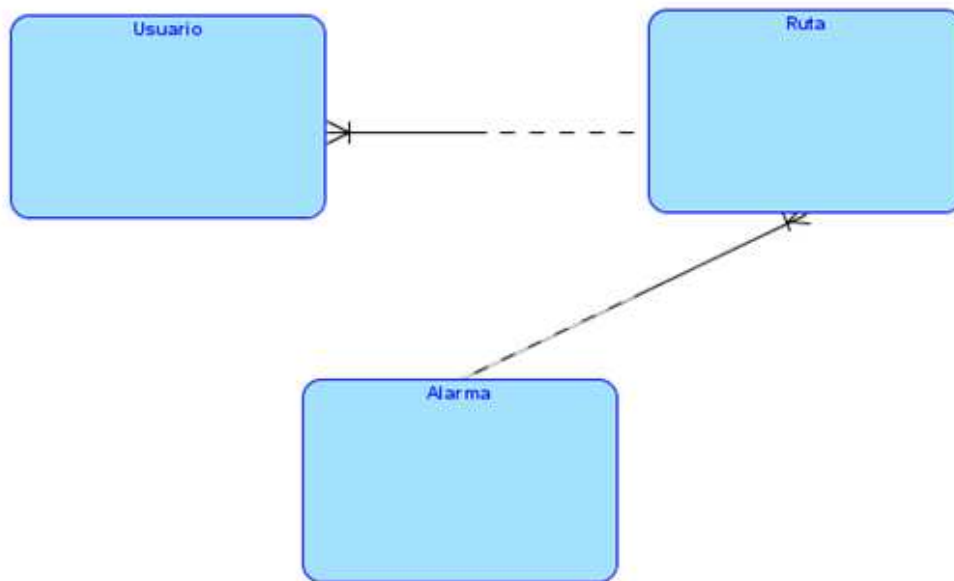


Figura 5.2.1 – Representación gráfica estructura lógica de información.

En esta fase de diseño, surge la primera decisión importante que debe ser tratada de forma minuciosa. Si se toma como referencia el modelo de información anterior (Figura 5.2.1) se asume que la información de la ruta estará contenida al completo dentro de la estructura de información “Ruta”.

Dado que una ruta, no es más que una sucesión de puntos de localización de GPS, surge una nueva propuesta de solución de modelo lógico de información.

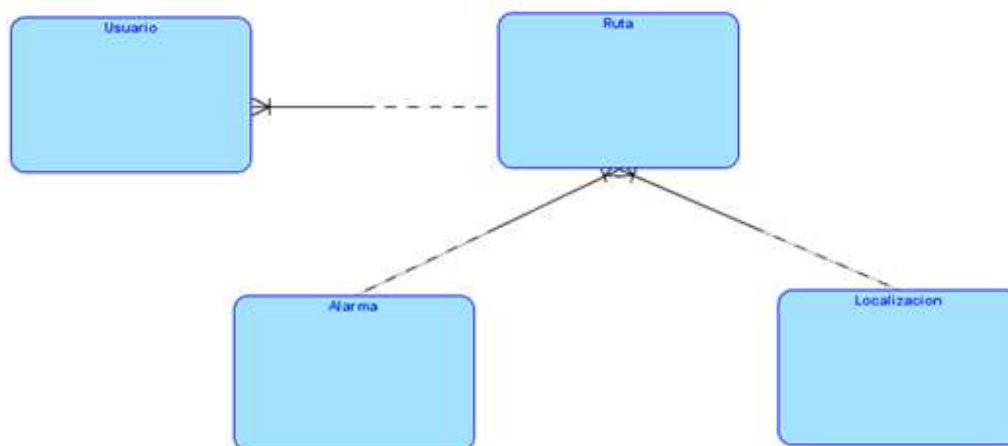


Figura 5.2.2 – Representación gráfica estructura lógica de información 2.

La diferencia entre el Modelo 1 y Modelo 2, sería la información relativa a los diferentes puntos de localización de los que se compone la ruta.

- Modelo 1 – Atributo dentro de la estructura de información Ruta.
- Modelo 2 – Atributo en una estructura de información independiente.

Finalmente se optará por la solución 2, que dota al modelo de mayor flexibilidad en la evolución de nuevas funcionalidades. No obstante, esta decisión podría quedar supeditada en un futuro en función de otros parámetros importantes como el número de usuario simultáneos realizando registros en el modelo.

Dado que en esta primera fase, no se han definido número máximo de usuarios simultáneos y al tratarse de una fase inicial en el que el número será reducido se optará por la propuesta de diseño número 2.

En cualquiera de las propuestas de solución, no se ha entrado a valorar los atributos que formarán parte de cada estructura de información, este detalle se definirá y diseñará en la estructura del modelo físico.

5.3. Estructura física del modelo

Partiendo del modelo de información lógica de la figura (Figura 5.2.2), se procede a definir las entidades y atributos de la estructura del modelo físico.

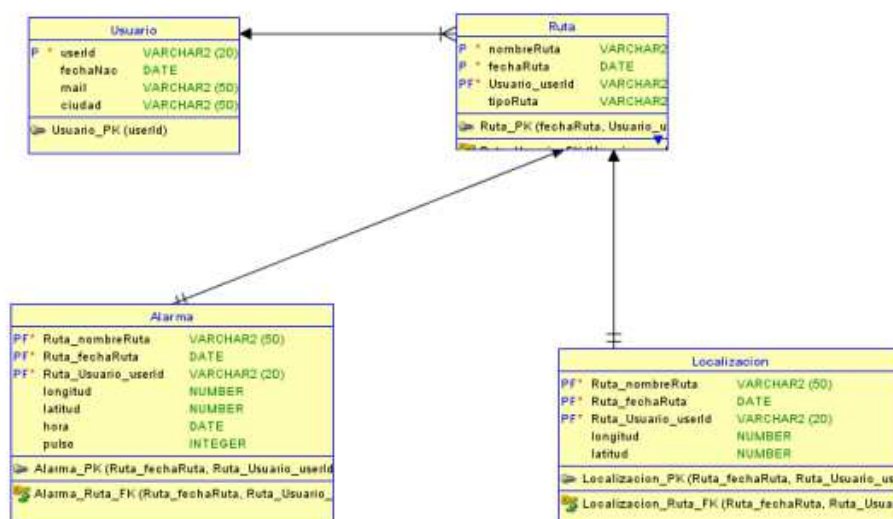


Figura 5.3.1 – Representación gráfica estructura Física

El modelo Físico se estructura según la figura (Figura 5.3.1), en el que se pueden identificar cada una de las entidades y relaciones de las que se compone el modelo.

Entidades:

- USUARIO: Tabla encargada de almacenar los usuarios registrados en la aplicación.

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 USERID	VARCHAR2 (20 BYTE)	No	(null)	1 (null)	
2 FECHANAC	DATE	Yes	(null)	2 (null)	
3 MAIL	VARCHAR2 (50 BYTE)	Yes	(null)	3 (null)	
4 CIUDAD	VARCHAR2 (50 BYTE)	Yes	(null)	4 (null)	

Todos los usuarios de la aplicación se identificarán por su USERID, identificador unívoco que será la PK de la tabla.

- RUTA: Tabla encargada de guardar la información de la ruta.

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 NOMBRRUTA	VARCHAR2(50 BYTE)	No	(null)	1 (null)	
2 FECHARUTA	DATE	No	(null)	2 (null)	
3 USUARIO_USERID	VARCHAR2(20 BYTE)	No	(null)	3 (null)	
4 TIPORUTA	VARCHAR2(2 BYTE)	Yes	(null)	4 (null)	

Las rutas se identificarán por su nombre de ruta, fecha de ruta y usuario de ruta, siendo esta la PK de la tabla.

Como atributo importante de la ruta, encontramos el tipo de ruta, información necesaria para el algoritmo “generación de alarmas”. Para el tipo de ruta se tendrán estos valores:

- *Run*
- *Bike*
- *Walk*

Cada tipo de ruta tendrá un peso determinado para calcular la distancia calculada aproxima entre dos puntos de localización y en función del valor calculado y el valor real obtenido se creará la alarma del sistema para una ruta.

- LOCALIZACIÓN: Tabla encargada de almacenar todos los puntos de localización GPS de una ruta.

Cada punto de localización se identificará por dos coordenadas, cada coordenada será utilizada para pintar el mapa todos los puntos de la ruta y dibujar la línea de la ruta en el mapa.

Las coordenadas estarán formadas por:

- Latitud
- Longitud

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 RUTA_NOMBRRUTA	VARCHAR2(50 BYTE)	No	(null)	1 (null)	
2 RUTA_FECHARUTA	DATE	No	(null)	2 (null)	
3 RUTA_USUARIO_USERID	VARCHAR2(20 BYTE)	No	(null)	3 (null)	
4 LONGITUD	NUMBER	Yes	(null)	4 (null)	
5 LATITUD	NUMBER	Yes	(null)	5 (null)	
6 HORA	DATE	Yes	(null)	6 (null)	
7 PULSO	NUMBER(38,0)	Yes	(null)	7 (null)	

:

Cada localización se identificará por su nombre de ruta, fecha de ruta y usuario de ruta, siendo esta la PK de la tabla.

- ALARMA: Tabla encargada de almacenar los diferentes avisos o alarmas que se generan en la ruta.

Al igual que la localización de la ruta, las alarmas generadas se identificarán por las coordenadas (latitud y longitud); además de la hora en la que se ha registrado la alarma y el pulso del usuario en el momento de la última localización.

La información del pulso es importante para descartar falsos positivos. Si el pulso es alto, el usuario seguirá en movimiento, mientras que si es bajo, podría tratarse de una parada no esperada.

❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1 RUTA_NOMBRERUTA	VARCHAR2(50 BYTE)	No	(null)	1 (null)	
2 RUTA_FECHARUTA	DATE	No	(null)	2 (null)	
3 RUTA_USUARIO_USERID	VARCHAR2(20 BYTE)	No	(null)	3 (null)	
4 LONGITUD	NUMBER	Yes	(null)	4 (null)	
5 LATITUD	NUMBER	Yes	(null)	5 (null)	

Cada alarma se identificará por su nombre de ruta, fecha de ruta y usuario de ruta, siendo esta la PK de la tabla.

5.4. Diagrama de secuencia

En este capítulo se representa el diagrama de secuencia UML y los diferentes flujos funcionales del sistema. Partiendo del modelado estático de los casos de uso, con el diagrama de secuencia se consigue un modelado dinámico, favoreciendo el entendimiento y comportamiento del sistema, así como las interacciones entre clases.

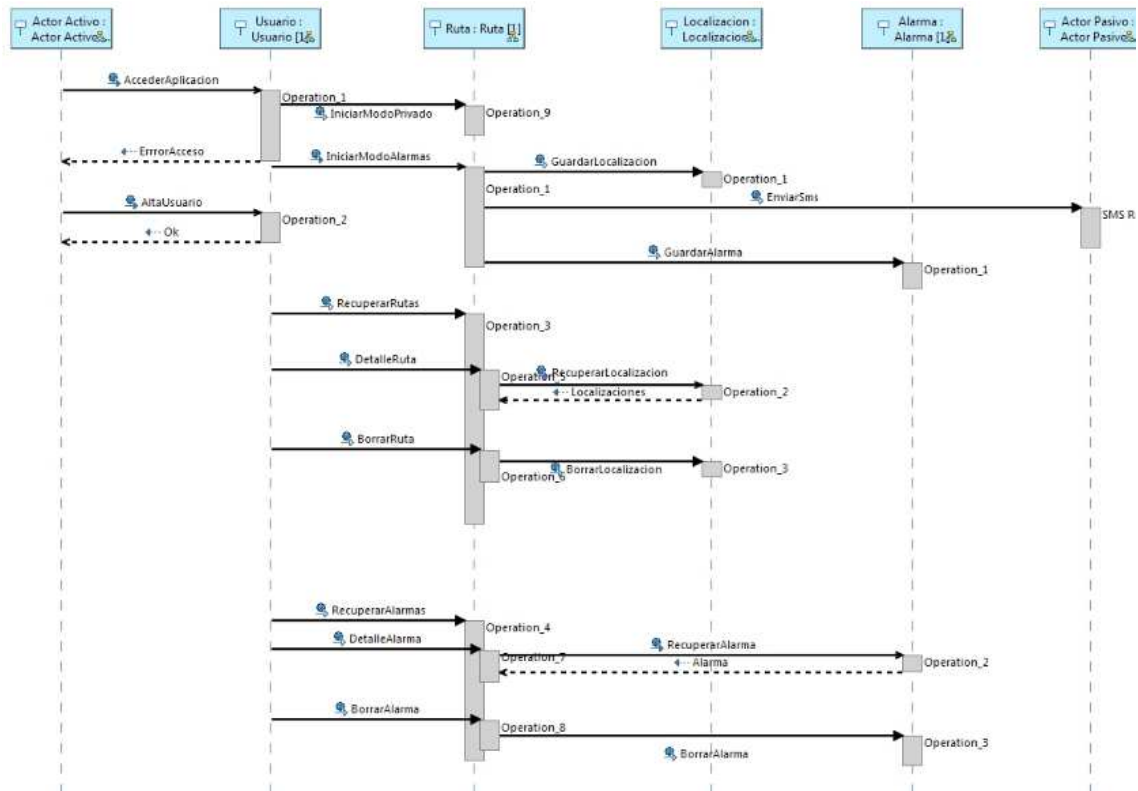


Figura 5.4.1 – Representación gráfica diagrama de secuencia.

Para mejorar la legibilidad del diagrama, no se han incluido las respuestas con resultado OK o KO de los diferentes métodos entre clases.

Capítulo 6. Implementación

Se procede a desarrollar e implementar la solución que debe cubrir los requisitos y diseño de los apartados anteriores y la arquitectura del sistema MVC. MVC es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Por lo tanto, el proyecto se estructura con una división de subdirectorios que permite independizar las tres capas.

- Subdirectorio 'src/main/resources', contiene todos los recursos utilizados por la aplicación y acceso a BBDD.
- Subdirectorio 'src/main/webapp', contiene todos los archivos que no sean código fuente Java, como archivos JSP y de configuración.
- Subdirectorio 'src/main/java' contiene todos los archivos fuente Java.
 - Archivos “Web” – Donde se definen los controladores de página.
 - Archivos “Domain” – Donde se definen los archivos de dominio de la aplicación.
 - Archivos “Service” – Donde se definen los diferentes algoritmos e implementaciones de la aplicación.
- El directorio 'target' donde se genera el archivo WAR que usa para almacenar y desplegar rápidamente nuestra aplicación.
- El fichero 'pom.xml' que contiene las dependencias Maven.
- Subdirectorio 'src/main/webapp/WEB-INF/views', en este directorio se almacenan todas las páginas de la aplicación accesibles desde el controlador.

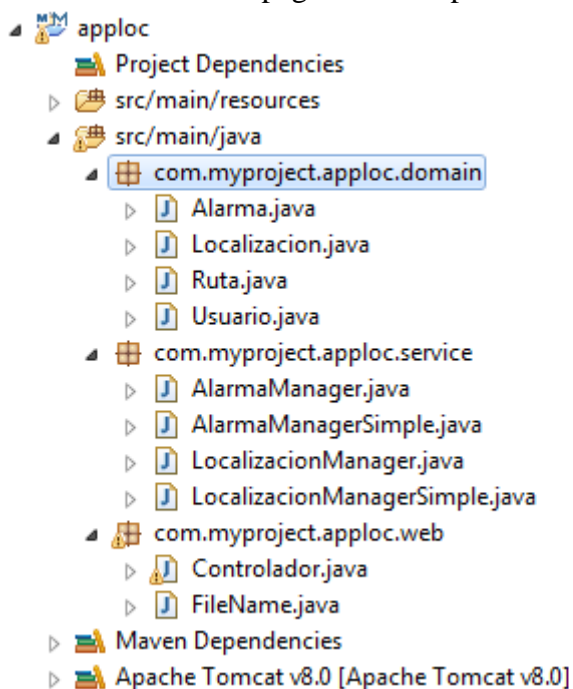


Figura 6.1– Representación gráfica estructura de proyecto

6.1. Clases de dominio

En este apartado se define la estructura de clases de dominio necesaria para el PFC, cada clase se corresponde con un “*Plain Old Java Object*” (POJO), clases java simples independientes de la arquitectura a utilizar.

Se corresponde con las clases que accederán posteriormente a la BBDD, se crea una clase de dominio para cada una de las tablas definidas e identificadas en apartados anteriores.

Por lo tanto, se definen las siguientes clases de dominio. Todo cambio que se produzca en cualquier de ellas (Ej. Cambio de gestor de BBDD), será transparente y no afectará la implementación de las clases de servicio y controladores del sistema.

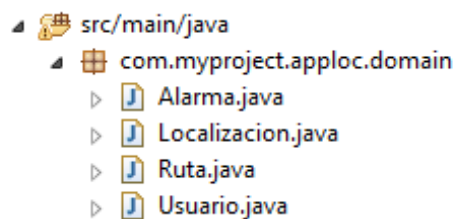


Figura 6.1.1 – Representación creación local clases de dominio.

Tomando como ejemplo la clase de dominio, Alarma, se aprecia la simplicidad de los métodos definidos, creándose operaciones de recuperación y modificación de cada elemento o columna que forma parte de la entidad Alarma.

```
package com.myproject.apploc.domain;
import java.util.Date;
public class Alarma {
    private double longitud;
    private double latitud;
    private Date fecha;

    public double getLongitud (){
        return longitud;
    }

    public double getLatitud (){
        return latitud;
    }
    public Date getfecha (){
        return fecha;
    }
    public void setLongitud (double longitud){
        this.longitud = longitud;
    }
    public void setLatitud (double latitud){
        this.latitud = latitud;
    }
    public void setFecha (Date fecha){
        this.fecha = fecha;
    }
}
```

Figura 6.1.2 – Representación gráfica código clase de dominio.

6.2. Clases de servicio

En este apartado se definen las clases de servicio, clase que se apoyan en las clases de dominio del apartado anterior pero en la que se incluye la implementación del software. Como sucede con las clases de dominio, se corresponde con POJOs independiente de la arquitectura.

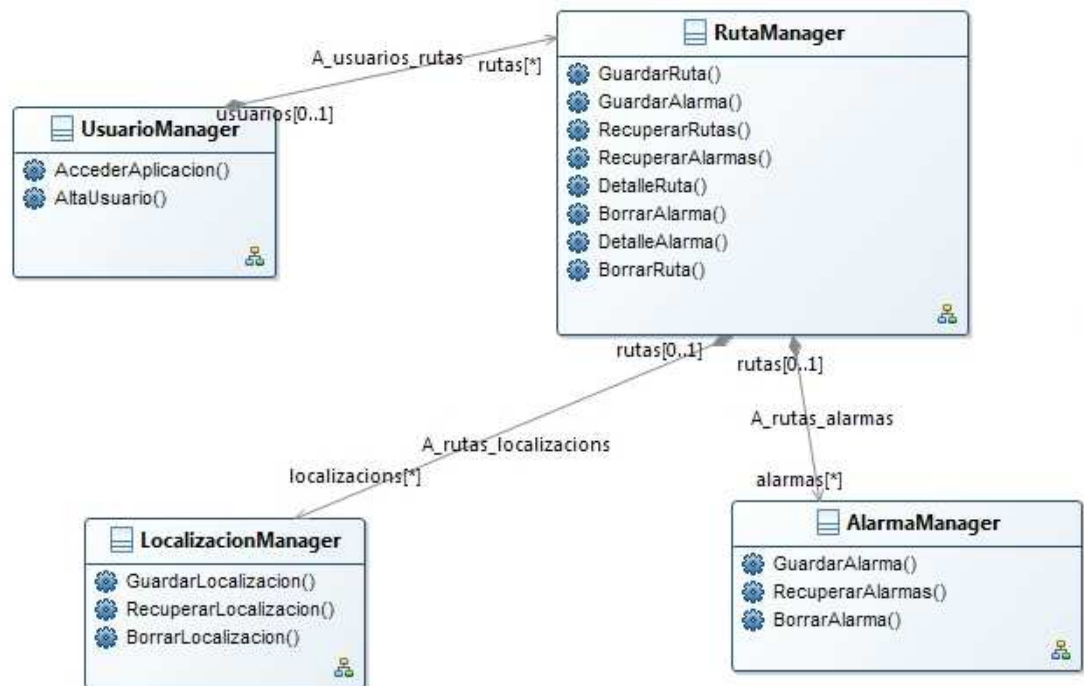


Figura 6.2.1 – Representación gráfica clases de servicio.

Las clases de servicio ofrecen las operaciones necesarias para que el controlador del sistema se comunique con la capa de presentación ofreciendo cada una de las operaciones del interfaz, tal y como se puede observar en el diagrama de secuencia.

Capítulo 7. Pruebas y validación

Se procede a validar la aplicación y la generación de las alarmas y rutas en función de un xml recibido. El resto de casos de prueba de los diferentes casos de uso y funcionalidades no se incluirán en la memoria dado que tienen una importancia menor frente al principal valor de la aplicación que es la generación de alarmas.

Se han identificado los siguientes casos de prueba para dar por certificada la aplicación.

- Prueba *MTB* con alarmas y falsos positivos descartados.
- Prueba *Running* con alarmas y sin falsos positivos.
- Prueba *Running* sin alarmas y falsos positivos descartados.
- Prueba *MTB* con múltiples alarmas y falsos positivos.
- Prueba *MTB* sin alarmas.

En los diferentes casos de prueba se validarán diferentes parámetros:

- Tipo de actividad.
 - *MTB y Running*. El ritmo en la actividad es totalmente diferente en cada tipo de actividad.
- Tipo de perfil.
 - Montaña y asfalto ciudad. En la montaña el ritmo es menos constante.
- Tipo de perfil usuario.
 - Usuarios experimentados vs usuarios principiantes. Los usuarios experimentados proporcionan ritmos constantes frente a usuarios principiantes que aportan ritmos irregulares.
- Duración.
 - Actividades cortas vs Actividades largas, actividades largas pueden generar falsos positivos porque están expuestos a trayectorias más largas que requieren paradas planificadas o pérdidas de recepción de satélites.

El objetivo principal de la pruebas, es que, en ninguno de los casos de prueba propuestos se generen falsos positivos.

7.1. Caso de prueba 1

Se clasifica el caso de prueba con los siguientes valores:

- Tipo Actividad: *MTB*.
- Tipo Perfil: Montaña.
- Tipo Usuario: Experimentado.
- Duración: Alta.

Se procede a analizar los resultados de un Track (Apéndice A) de duración aproximada, 3h15min, donde se identificaron 3 paradas durante una actividad de MTB y la calidad de la señal es óptima.

Se obtienen los siguientes resultados:

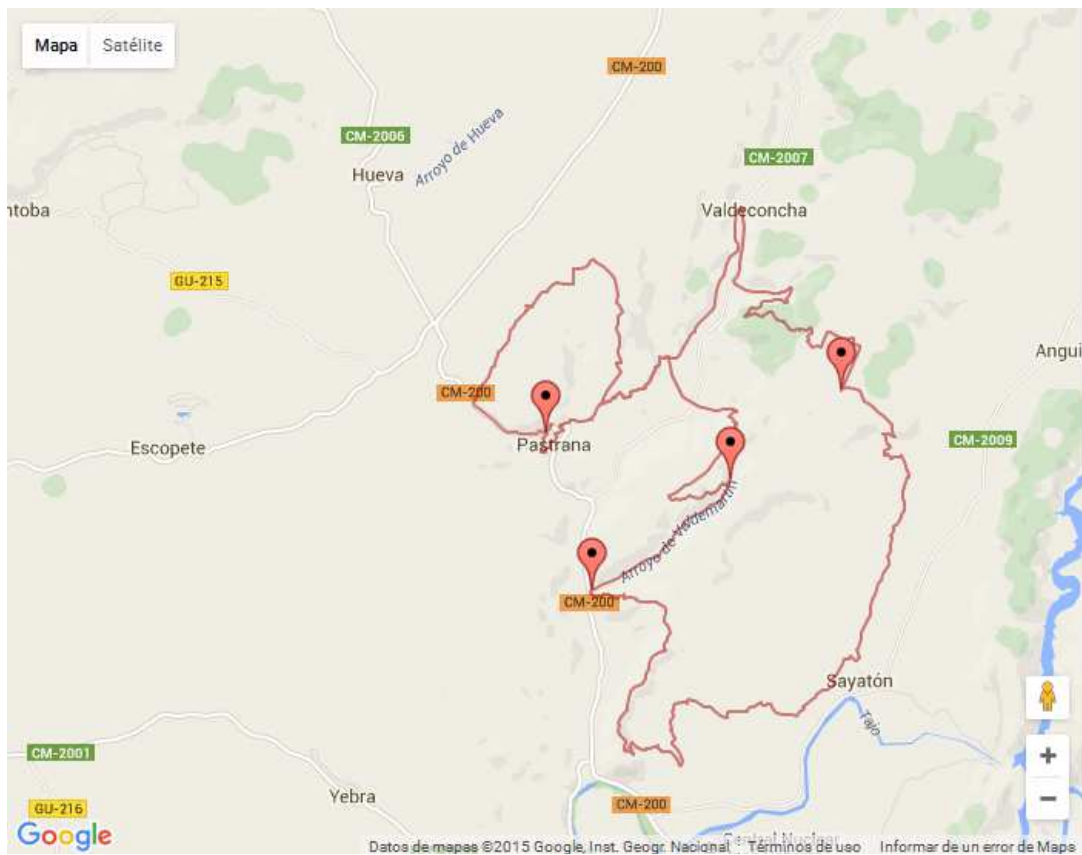


Figura 7.1.1 – Resultados alarma prueba 1.

Las pruebas son satisfactorias y se coteja el resultado con la velocidad registradas para ese mismo *Track* en la aplicación Garmin Connect.



Figura 7.1.2 – Información velocidad alarma prueba 1.

Como se puede observar con la gráfica de velocidad (Fuente Garmin Connect), se aprecian claramente 4 paradas, 3 de ellas identificadas como paradas reales y una de ellas no contemplada.

Puntos a destacar.

El desarrollo del algoritmo, es capaz de descartar la última parada, posiblemente por una señal de GPS defectuosa, en la primeras versiones de algoritmo, no se conseguía identificar como un falso positivo.

Puntos a mejorar.

El desarrollo del algoritmo, no es capaz de diferenciar que 2 de las paradas se corresponden a avituallamientos, paradas planificadas.

7.2. Caso de prueba 2

Se clasifica el caso de prueba con los siguientes valores:

- Tipo Actividad: *Running*.
- Tipo Perfil: Montaña.
- Tipo Usuario: Experimentado.
- Duración: Alta.

Se procede a analizar los resultados de un *Track* (Apéndice B) de duración aproximada de 2h donde se identificaron 4 paradas durante una actividad de *Running* de montaña y la calidad de la señal es óptima.

Se obtienen los siguientes resultados:

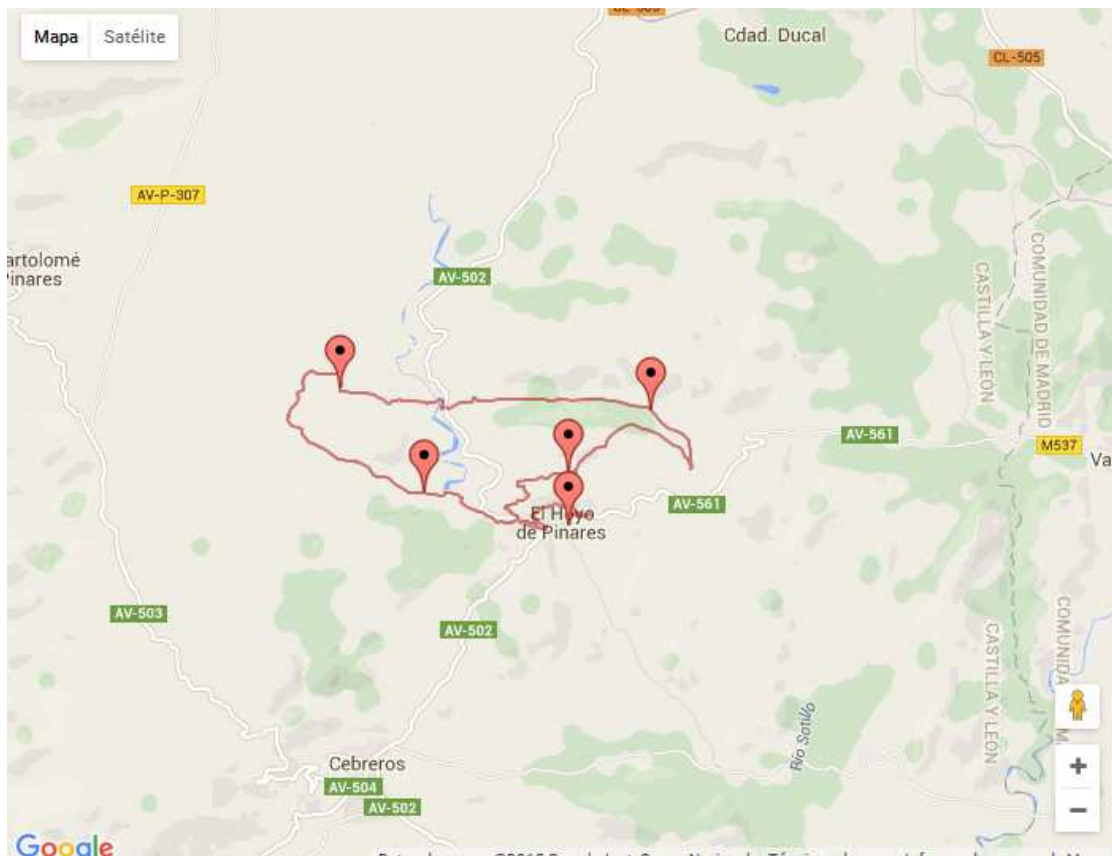


Figura 7.2.1 – Resultados alarma prueba 2.

Las pruebas son satisfactorias y se coteja el resultado con la velocidad registradas para ese mismo *Track* en la aplicación Garmin Connect.



Figura 7.2.2 – Información velocidad alarma prueba 2.

Como se puede observar con la gráfica de velocidad (Fuente Garmin Connect), se aprecian claramente 4 paradas. En este caso de prueba, todas las paradas identificadas como alarmas son reales y no se produce ningún falso positivo.

Puntos a destacar.

En este caso de prueba, a pesar de tratarse de una prueba de montaña, la velocidad en una prueba de *Running* es más constante; por lo que la identificación de la alarma por parada es más sencilla.

Puntos a mejorar.

El desarrollo del algoritmo, no es capaz de diferenciar que las 4 paradas se corresponden a avituallamientos, paradas planificadas.

7.3. Caso de prueba 3

Se clasifica el caso de prueba con los siguientes valores:

- Tipo Actividad: *Running*.
- Tipo Perfil: Asfalto.
- Tipo Usuario: Experimentado.
- Duración: Alta.

Se procede a analizar los resultados de un *Track* (Apéndice C) de duración aproximada de 3h donde se identificaron paradas durante una actividad de *Running* y la calidad de la señal es óptima.

Se obtienen los siguientes resultados:

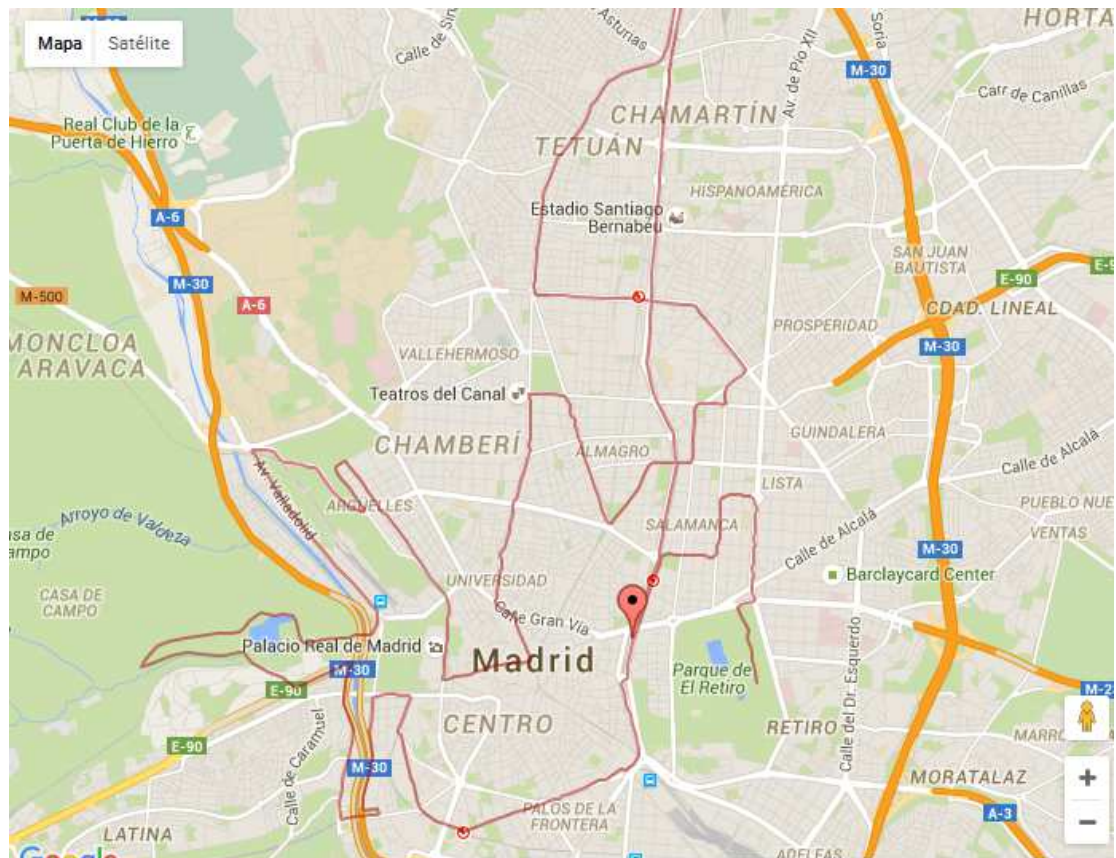


Figura 7.3.1 – Resultados alarma prueba 3.

Las pruebas son satisfactorias y se coteja el resultado con la velocidad registradas para ese mismo *Track* en la aplicación Garmin Connect.



Figura 7.3.2 – Información velocidad alarma prueba 3.

Como se puede observar con la gráfica de velocidad (Fuente Garmin Connect), no se aprecian descensos de la velocidad, exceptuándose en la parte final, donde la velocidad se reduce sin llegar a producirse parada.

Puntos a destacar.

En este caso de prueba, al tratarse de una prueba de asfalto donde el ritmo es constante el tratamiento de alarmas es mucho más sencillo; no obstante, se identifica una nueva derivada que no se había identificado, la altura o pendiente. Si procedemos a analizar este mismo *Track* desde Garmin Connect con la información de altura:



Figura 7.3.3 – Información velocidad alarma prueba 3 - altura.

Como se puede observar en la figura (7.2.3 Fuente Garmin Connect) existe un descenso considerable en la velocidad debido a la pendiente. En esta fase del proyecto, se puede dar por ok los resultados obtenidos, pero para fases posteriores se debería contemplar los parámetros de altura en el algoritmo de alertas.

Puntos a mejorar.

Incluir en el algoritmo parámetros de altura.

7.4. Caso de prueba 4

Se clasifica el caso de prueba con los siguientes valores:

- Tipo Actividad: *MTB*.
- Tipo Perfil: Montaña.
- Tipo Usuario: Principiante.
- Duración: Alta.

Se procede a analizar los resultados de un *Track* (Apéndice D) de duración aproximada de 3h donde se identificaron paradas durante una actividad de *MTB* y la calidad de la señal es óptima.

Se obtienen los siguientes resultados:

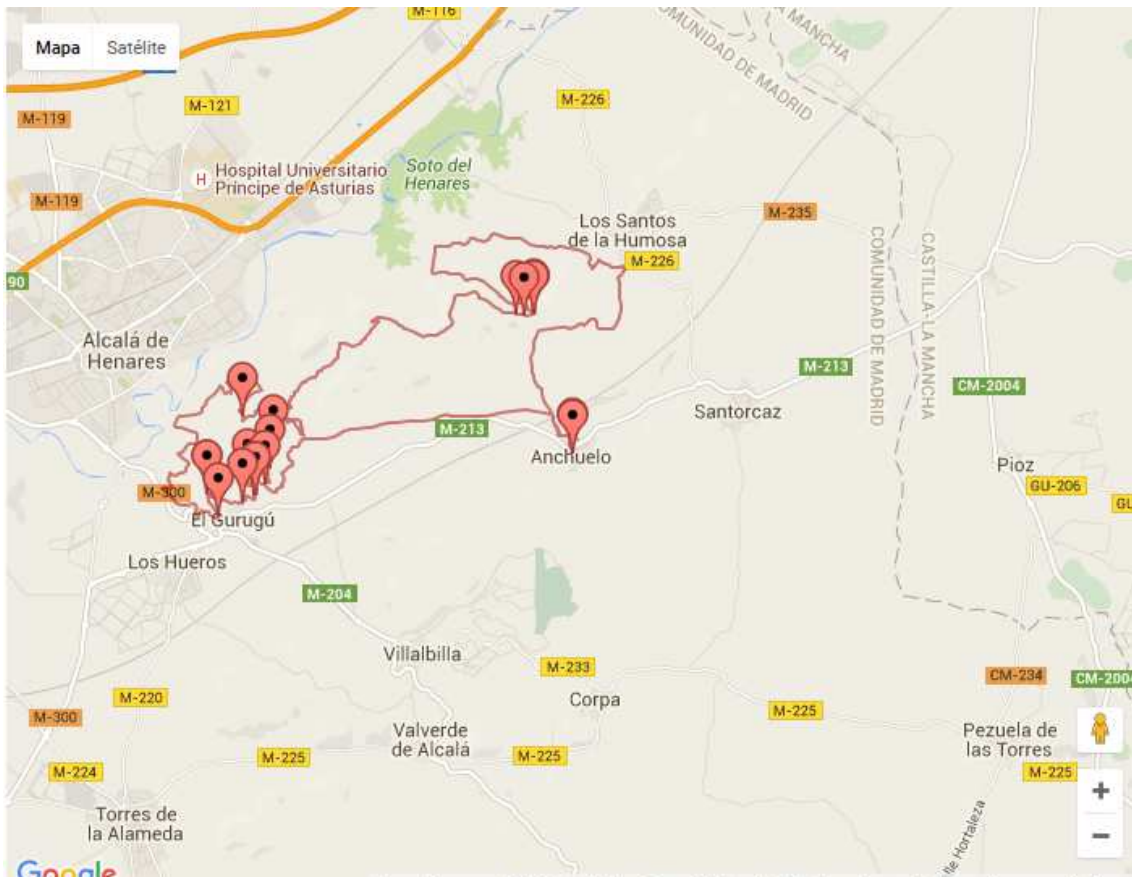


Figura 7.4.1 – Resultados alarma prueba 4.

Las pruebas en este caso, son satisfactorias y se coteja el resultado con la velocidad registradas para ese mismo *Track* en la aplicación Garmin Connect. No obstante, se producen muchas más alarmas de las previstas para una ruta de duración aproximada de 3h.

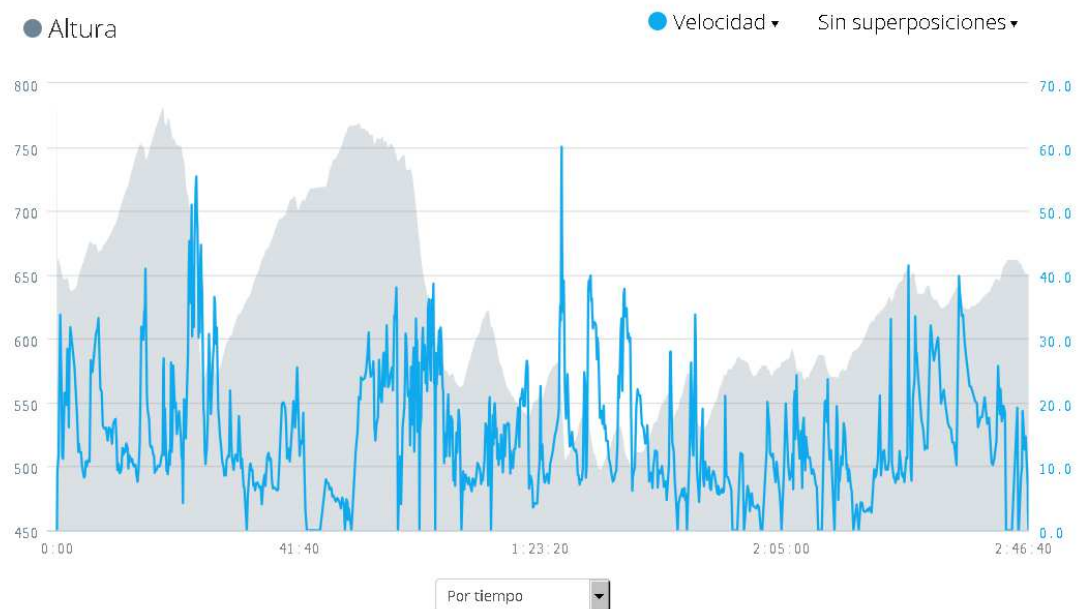


Figura 7.4.2 – Información velocidad alarma prueba 4.

Como se puede observar con la gráfica de velocidad (Fuente Garmin Connect), se observan descensos continuados en la velocidad, ocasionados por paradas reales o por desniveles importantes de pendiente.

Puntos a destacar.

Se monitorizan, como en los casos de prueba anteriores, todas las alarmas correctamente.

Puntos a mejorar.

Incluir en el algoritmo parámetros de altura y perfil de usuario que permita generar únicamente los avisos reales. En este caso de prueba, es el primero en el que nos encontraríamos con falsos positivos.

7.5. Caso de prueba 5

Se clasifica el caso de prueba con los siguientes valores:

- Tipo Actividad: *MTB*.
- Tipo Perfil: Asfalto.
- Tipo Usuario: Principiante.
- Duración: Baja.

Se procede a analizar los resultados de un *Track* (Apéndice D) de duración aproximada de 1h donde no se identificaron paradas durante una actividad de *MTB* y perfil plano y asfalto..

Se obtienen los siguientes resultados:

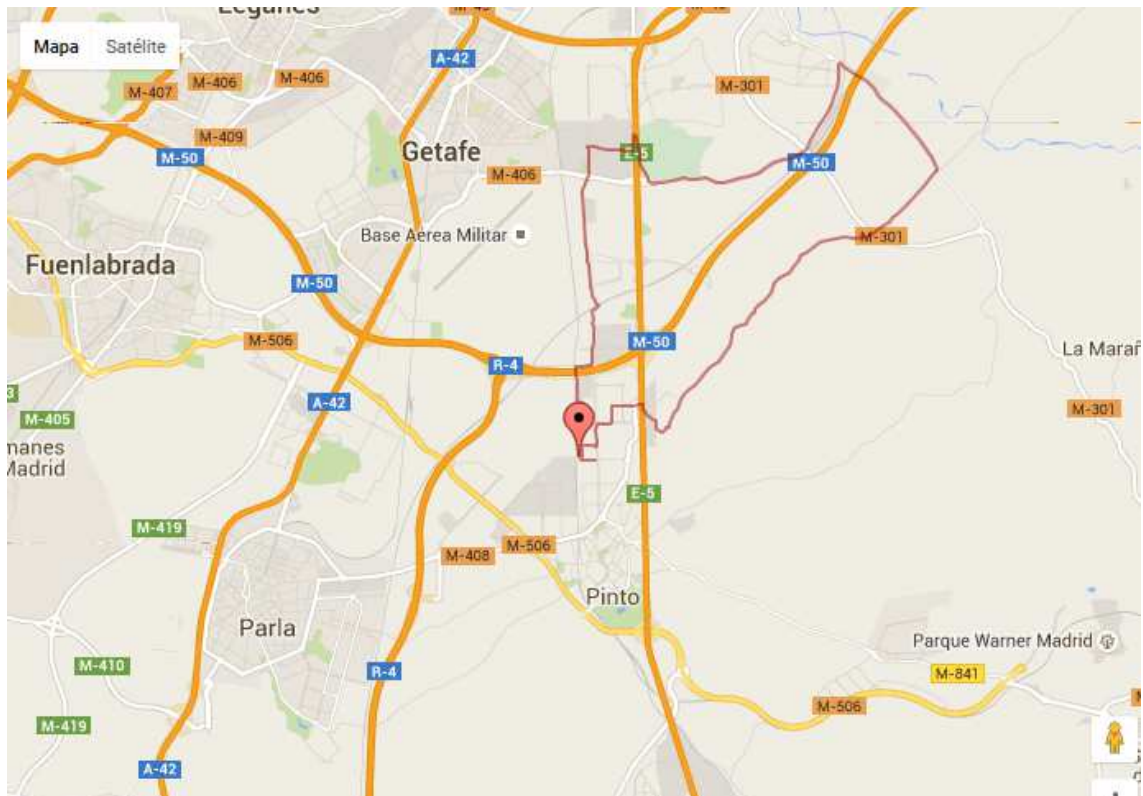


Figura 7.5.1 – Resultados alarma prueba 5.

Las pruebas en este caso, son satisfactorias y se coteja el resultado con la velocidad registradas para ese mismo *Track* en la aplicación Garmin Connect.



Figura 7.5.2 – Información velocidad alarma prueba 5.

Como se puede observar en la gráfica de velocidad (Garmin Connect), aunque se aprecia descenso de velocidad, dado que el desnivel es plano, no se generan alarmas ni falsos positivos.

Puntos a destacar.

Resultados esperados.

Puntos a mejorar.

No se contemplan, dado que las variables de las pruebas eran favorables:

- Duración corta.
- Desnivel plano y asfalto.
- MTB

Capítulo 8. Planificación y presupuesto

En este capítulo se desarrollan dos apartados. En el primero de ellos se detalla la planificación realizada para el presente trabajo, mientras que en el segundo se incluye un presupuesto del mismo.

8.1. Planificación del proyecto

A continuación, se indican cada una de esas fases que se han ido siguiendo, detallando, a su vez, las principales tareas que componen cada una de ellas:

- **Planteamiento**
 - ✓ Definición de objetivos y alcance del proyecto.
- **Estado del arte**
 - ✓ Identificar sistemas similares, características comunes y diferencias.
 - ✓ Entornos de desarrollo integrados y lenguaje de programación.
 - ✓ Sistema de gestión de bases de datos.
 - ✓ Servidores web.
 - ✓ Sistemas de localización de señal y posicionamiento.
 - ✓ Algoritmo de distancia entre puntos.
- **Análisis**
 - ✓ Definición de requisitos funcionales y no funcionales.
 - ✓ Casos de uso y trazabilidad con requisitos.
- **Diseño**
 - ✓ Arquitectura de sistema.
 - ✓ Evaluación distintas alternativas de diseño lógico.
 - ✓ Definición diseño final.
- **Implementación**
 - ✓ Creación de clases de dominio.
 - ✓ Creación de clases de servicio.
 - ✓ Integración de componentes y test.
- **Pruebas**
 - ✓ Realización de pruebas y validación de resultados.
- **Documentación del proyecto**
 - ✓ Memoria del proyecto.
 - ✓ Presentación PowerPoint del proyecto.

Después de indicar las principales fases y tareas identificadas para el desarrollo del PFC, se incluye la secuencia y temporización de cada una de ellas mediante un diagrama de Gantt, el cual se muestra a continuación:

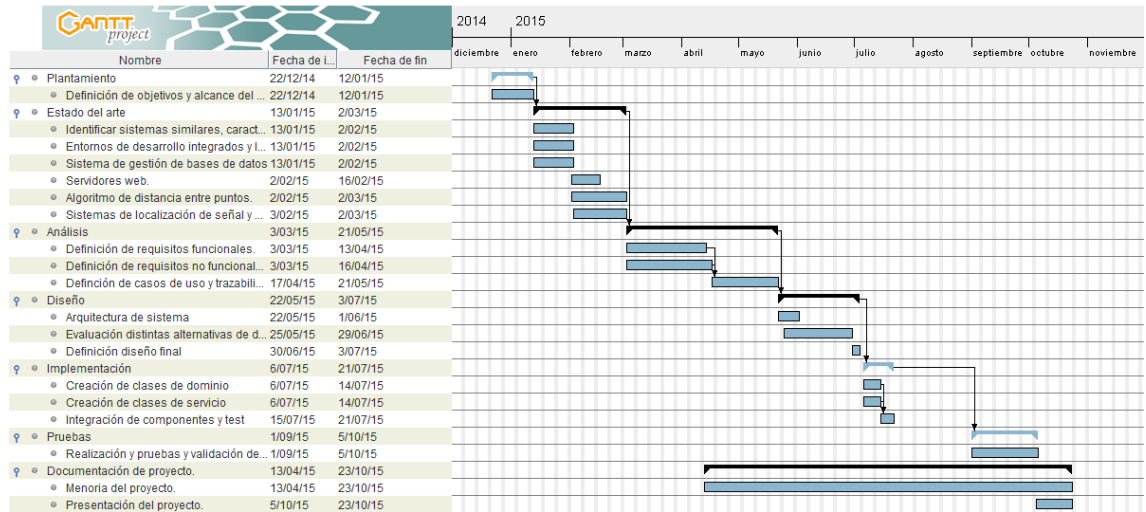


Figura 8.1.1 – Diagrama Gantt planificación de proyecto.

8.2. Presupuesto del proyecto

En este apartado se elabora el presupuesto del PFC. Se ha tenido en cuenta los diferentes gastos acometidos durante la realización del mismo, costes materiales, así como el tiempo dedicado a la realización del trabajo para calcular los costes de recursos humanos, cabe destacar que la unidad monetaria utilizada es el Euro (€). Los costes se desglosan en los siguientes tipos:

- Costes de recursos humanos
- Costes de material
- Costes indirectos

8.2.1. Costes de recursos humanos

Para evaluar los costes de recursos humanos, se requiere en primer lugar contabilizar el número de horas dedicadas a la realización del mismo durante los meses de trabajo detallados en la planificación del apartado anterior.

Mes	Días	Horas	Total
Diciembre	8	3	24
Enero	8	3	24
Febrero	8	3	24
Marzo	16	3	48
Abril	16	3	48
Mayo	16	3	48
Junio	12	3	36
Julio	10	3	30
Agosto	0	0	0
Septiembre	20	5	100
Octubre	20	5	100
Total			482

Tabla 8.2.1.1 – Inventario de horas dedicadas.

En la Tabla 8.2.1.1 se puede observar que las horas totales dedicadas al proyecto, con un total de 482h.

Una vez que han sido contabilizadas las horas totales dedicadas al proyecto, se procede a calcular los costes asociados a los recursos humanos. Para ello se tendrá en cuenta al autor del Proyecto Fin de Carrera, Alberto Herranz, como único recurso a contabilizar en los costes de recursos humanos. En la *Tabla 8.2.1*, que se detalla a continuación, se muestran los costes asociados al personal, teniendo en cuenta las horas totales dedicadas calculadas en la tabla anterior y asumiendo un coste de 30€/hora de un Ingeniero Técnico.

Recurso	Categoría	Coste (€/Hora)	Horas dedicadas	Total (€)
Alberto Herranz	Ingeniero Técnico	30,00 €	482	14.460,00 €

Tabla 8.2.1.1 – Total costes recursos humanos.

8.2.2. Costes de material

A la hora de realizar el cálculo total asociado a los recursos materiales, se ha desglosado el coste por cada elemento utilizado, teniendo en cuenta la dedicación al proyecto, el precio de cada herramienta, su porcentaje de uso y el período de amortización. El cálculo se efectuará mediante la siguiente fórmula:

$$\text{Coste} = \text{Precio elemento} * \text{Porcentaje de uso} * \frac{\text{Número meses dedicación}}{\text{Período de depreciación}}$$

A continuación, se incluye una tabla en la que se desglosa el coste de cada material, distinguiendo entre materiales hardware y software:

Descripción material	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable ^{d)}
HP EliteBook 8440p	420,00	100	10	60	70,00
Windows 7 Enterprise	190,00	100	10	60	31,67
Microsoft office 2010	210,00	100	10	60	35,00
Total					136,67

Tabla 8.2.2.1 – Total costes material.

Como se puede comprobar en la *Tabla 8.2.2.1*, los costes de material ascienden a 136,67€.

8.2.3. Costes indirectos

Por lo que se refiere a los costes indirectos del proyecto, se ha tenido en cuenta únicamente el gasto de la conexión a internet, ADSL.

Descripción	Coste/mes (€)	Meses dedicación	Coste (€)
ADSL	30	10	300
TOTAL			300,00 €

Tabla 8.2.3.1 – Total costes indirectos.

8.2.4. Resumen de costes

Por último, se incluye una tabla final en la que se suman los tres tipos de coste desglosados.

Descripción	Coste (€)
Costes totales de recursos humanos	14.460,00 €
Costes totales de material	136,67 €
Costes indirectos totales	300,00 €
Importe total proyecto	14.896,67 €

Tabla 8.2.3.1 – Total costes PFC.

Capítulo 9. Conclusiones y trabajo futuro

Se procede a desarrollar el apartado más importante del PFC, como son las conclusiones que se desprenden del trabajo realizado y grado de satisfacción sobre los objetivos cumplidos; así cómo, el trabajo futuro que se podría realizar sobre el mismo PFC.

La idea inicial del PFC era la creación de una aplicación Web que gestionara carreras, corredores, entrenamientos y todo tipo de eventos relacionados con el mundo del “*running*” y el ciclismo, basándose en:

- Eclipse/Java/Spring/Maven
- BBDD Oracle.

A medida que se empieza a definir la solución y estrategia a seguir, esta idea inicial pierde interés dado el gran número de aplicaciones existentes en la actualidad que realizan ese tipo de gestiones. Por lo tanto, la primera decisión importante, considerada como acertada, fue el cambio sobre el objetivo inicial del PFC dentro de la misma temática y tecnología, por una propuesta más innovadora y atractiva, pero a su vez más arriesgada desde el punto de vista técnico y objetivos a cumplir. Además, se debía contar con el agravante adicional del tiempo perdido en la idea original del proyecto, en resumen, se echaba el tiempo encima.

Una vez se decide el cambio en el objetivo principal del PFC, por el sistema de alertas *Non-Stop*, se procede a realizar un análisis de viabilidad teniendo en cuenta los nuevos factores:

- Tiempo para desarrollar el proyecto.
 - Tiempo reducido.
- Viabilidad tecnológica para ejecutar la nueva idea.
 - Señal GPS.
 - Disponibilidad en dispositivos móviles.
 - Dispositivos deportivos como Garmin.
 - Visualizar en un mapa las localizaciones de alarma generadas.
 - API – Google Maps.

La respuesta a estas cuestiones, aun siendo satisfactoria, supone un esfuerzo mayor en el análisis de tecnologías a emplear y el tiempo destinado a cada una de ellas. Por lo tanto, se tiene que conseguir una solución sencilla pero con potencial suficiente para desarrollarse y explotarse en un trabajo futuro.

De los objetivos definidos en el inicio del proyecto, se presentaron los siguientes:

Sistema Servidor. Receptor de señal GPS y emisor de alarma. Almacena la información en BBDD para explotación de ésta en un momento posterior o de manera on-line.

Objetivo	Cumplimiento
Aplicación Web-Maven	OK
BBDD – Oracle	OK
Presentación Web	Pendiente.

Parte más tecnológica del proyecto, empleando tecnología de proyectos WEB con Spring y gestión de dependencias Maven.

Sistema móvil. Emisor señal GPS

Objetivo	Cumplimiento
Aplicación Android	Pendiente.

Para evitar que el proyecto se parara y no pudiera cumplir la parte principal de éste, se opta por una solución temporal. Recuperar un Track de GPS previamente almacenado y realizar una simulación del sistema de alarmas. De esta forma se consigue el objetivo principal del proyecto como es la generación de las alarmas y localizaciones en el mapa de los puntos conflictivos.

Software - sistema de alarma. Ubicado en el sistema servidor.

Objetivo	Cumplimiento
Generar alarmas por parada.	OK
Visualizar alarmas en mapa	OK

Tras las pruebas realizadas, se consigue implementar un algoritmo de generación de alarmas en función de dos parámetros:

- Distancia esperada entre dos puntos: Algoritmo de “Haversine”.
- Distancia recorrida real entre dos puntos.

En este punto, se centraliza una parte importante de los esfuerzos del proyecto, se generan Tracks de GPS intencionados. Se fuerzan las paradas para analizar la información con el Algoritmo implementado y poder sacar conclusiones sobre la información aportada por los tracks generados.

- La señal GPS, se recibe con una periodicidad, según la calidad de la señal, entre 1 a 6 segundos; por lo tanto, es probable que se generan puntos consecutivos continuos en la detección de una parada. Una misma alarma está compuesta por una sucesión de puntos de localización GPS consecutivos.
- La señal GPS, puede contar con errores en la recepción de la señal; por lo tanto, se generaban falsos positivos. Una generación de alarma sin confirmación de puntos sucesivos, se debe considerar como un falso positivo.

Teniendo en cuenta estos factores, se consigue con cierta precisión, agrupar en un único punto la localización de una alarma y reducir el número de falsos positivos tal y como se puede ver en el apartado de demostración de resultados.

Uno de los puntos a mejorar del algoritmo, se produce cuando se mezclan dos derivadas, como son tipo de perfil montaña y tipo de usuario principiante (caso de prueba 4); por lo que se debe tener en cuenta como trabajo futuro.

Una vez analizados la consecución de los objetivos propuestos, se podría concluir que se han cumplido las expectativas generadas de forma satisfactoria, teniendo en cuenta los objetivos cumplidos y que se ha creado un sistema con una base de gran utilidad y potencial para un trabajo futuro atractivo e innovador.

- Objetivos técnicos – Aplicación Web - MAVEN con persistencia de BBDD Oracle.
- Objetivo funcional – Algoritmo generador de alarmas, con pruebas satisfactorias.

A continuación se pasa a detallar el trabajo futuro que se puede realizar a partir de la base creada en el PFC ‘Non-Stop’

- Trabajar con señal de uso civil como Galileo. El sistema de navegación europeo por satélite cada vez está más cerca de ser una realidad, ya dispone de 10 satélites en órbita.
La ventaja que dispone con respecto a la señal GPS estadounidense, de carácter militar, es su uso civil. Por lo tanto, no puede ser alterada la orientación de los satélites de forma premeditada generándose localizaciones erróneas.
- Trabajar con parámetros de altura que permitan identificar ritmos muy lentos por pendientes pronunciadas.
- Evolucionar aquellos puntos pendientes que no se han podido cubrir en el PFC y que eran objetivos iniciales.

Capítulo 10. Bibliografía y referencias

- [REF 1] Instituto Nacional de Estadística - <http://www.ine.es/>
- [REF 2] Consejo superior de deportes - www.csd.gob.es
- [REF 3] Eclipse - <https://www.eclipse.org/>
- [REF 4] Java - <https://www.java.com/es/>
- [REF 5] Spring - <http://projects.spring.io/spring-framework/>
- [REF 6] Oracle - <http://www.oracle.com/es>
- [REF 7] IBM - <http://www.ibm.com/es/es/>
- [REF 8] Apache Tomcat - <http://tomcat.apache.org/index.html>
- [REF 9] Spring - <https://spring.io/>
- [REF 10] API Google Maps - https://www.google.com/intx/es_es/work/mapsearch/products/mapsapi.html
- [REF 11] Open Street Map - <http://www.openstreetmap.es/>
- [REF 12] Endomondo - <https://www.endomondo.com/>
- [REF 13] Runtastic - <https://www.runtastic.com/es>
- [REF 14] Garmin Connect - <https://connect.garmin.com/es-ES/>
- [REF 15] Alpify - <http://www.alpify.com/es/aplicacion/>

Apéndices

Apéndice A. Vocabulario común y acrónimos.

Término	Definición
PFC	Proyecto fin de carrera
PC	Ordenar o computador personal
CPU	Unidad central de procesamiento
DDS	Documento de definición de sistema
INE	Instituto nacional de Estadística
CSD	Consejo Superior de Deportes
IDE	Entorno de desarrollo integrado
SMS	Servicio corto de mensajes. Short Message Service
Open Source	Código abierto y desarrollado libremente
JVM	Java Virtual Machine
JRE	Java Runtime Environment
JDK	Java Development Kit
API	Application Programming Interface
IDE	Entorno de Desarrollo Integrado
Servlet	Clase Java utilizada para ampliar las capacidades de un servidor
JSP	Java Server Page
BBDD	Bases de datos
JPA	Java Persistence API
UTF-8	Formato de codificación de caracteres
BBDD	Bases de Datos
SGBD	Sistema Gestor Base de Datos
GNSS	Sistema Global de Navegación por Satélite